FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Dendro Keywords: Supporting data description through information extraction

Cláudio Luís de Sousa Monteiro

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Carla Teixeira Lopes

Second Supervisor: João Rocha Silva

July 31, 2008

Dendro Keywords: Supporting data description through information extraction

Cláudio Luís de Sousa Monteiro

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: João Pedro Mendes Moreira
External Examiner: Ricardo Campos
Supervisor: Carla Teixeira Lopes
July 31, 2008

Abstract

Research and data are inseparable concepts since data is essential for research. Research data includes data created and collected with the purpose of producing results related to research.

Research data management relates to the management of data from its entry in the research cycle to the evaluation of results and its aim is to make the research process more efficient. Proper management helps to ensure the usability and availability of the data, improves the data retrieval process and allows data reuse while reducing its loss. Although important, it is a complex subject in which not every researcher has in-depth knowledge about, often needing help from data management professionals for whom there are not always funds available. Also the amount of time required when not every benefit is short term makes them rather focus on the work. This usually makes the researcher store data without appropriate descriptions. The quantity and diversity of metadata schemas also difficult the process of choosing data descriptors, which is essential for the preservation of data. On one hand, generic data descriptors can be used for interoperability and on the other hand it may be necessary the use of specific terms from a domain to allow a more detailed description of the data.

There are tools that help the researcher manage their data such as CKAN and Zenodo, but these tools often focus on finished work. Dendro is a research data management platform, that focuses on the early stages of research and was designed to offer researchers a collaborative environment to store and describe their datasets using the most adequate metadata descriptors from a set of existing ontologies. Currently in Dendro, an ontology is created after a meeting between the curator, who gives an introduction about data management, and the domain experts who provide the scientific context. This will allow the curator to create an ontology oriented to the researcher's needs.

The goal of our work is to develop a tool to be used in the ontology building process by helping curators with the extraction and visualization of key concepts in the research documents or publications by the group.

The first step consisted in evaluating different techniques used in ontology learning and choosing those that revealed itself more efficient. The selected approaches were then combined and introduced to Dendro in an implementation testbed. The resulting approach is comprised of a Natural Language Processing tool, different term extraction methods and DBpedia and LOV integration in order to provide concepts and descriptors for the previously extracted terms.

In order to evaluate the proposed approaches we have done both an automatic and a manual evaluation. For the automatic evaluation we selected 3 case studies, namely 3 different ontologies created within Dendro with the corresponding materials used during their creation. The manual evaluation was done with the assistance of the curators that provided those ontologies. They were given a selection of scientific documents to be used as input and were guided through the different phases of the tool while providing feedback accordingly.

We have developed a tool that was able to find the majority of the expected concepts. The opinions of the curators that have evaluated the tool were also favorable, as they agree that a tool

like Dendro Keywords is a good addition to their work.

Resumo

Investigação e dados são termos inseparáveis, uma vez que os dados são essenciais para a investigação. Dados de investigação são os dados que são criados e colecionados com o intuito de produzir resultados relacionados com o trabalho.

A gestão de dados de investigação está relacionada com a gestão dos dados desde a sua entrada no ciclo da investigação até à avaliação de resultados e o seu objetivo é tornar a investigação mais eficiente. Uma boa gestão ajuda a assegurar a usabilidade e disponibilidade dos dados, melhora o processo de recolha e permite a reutilização de dados enquanto reduz sua perda. Mesmo que importante, continua a ser um assunto complexo, uma vez que, nem todos os investigadores possuem conhecimentos aprofundados de gestão de dados, o que os leva a precisar muitas vezes de ajuda de profissionais, para os quais nem sempre existem fundos. Também a quantidade de tempo necessário quando nem todos os benefícios são a curto prazo leva-os a preferirem focar-se no trabalho. Isto leva o investigador a armazenar muitas vezes os seus dados sem descrição. A quantidade e diversidade de esquemas de metadados também dificulta o processo de escolha de descritores, que são essenciais para a descrição de dados. Por um lado, descritores genéricos podem ser utilizados para interoperabilidade e por outro podem ser necessários termos específicos de um certo domínio para permitir uma melhor descrição dos dados.

Existem ferramentas que ajudam o investigador a gerir os seus dados, tais como, CKAN e Zenodo, mas estas ferramentas muitas vezes focam-se no trabalho já concluído. O Dendro é uma plataforma de gestão de dados, que ao contrário das anteriores, se foca nos instantes iniciais da investigação e foi desenhado de forma a oferecer aos investigadores um ambiente colaborativo para armazenar e descrever os seus dados, usando os descritores de metadados mais adequados de um conjunto de ontologias existentes. Neste momento no Dendro, uma ontologia é criada após uma reunião entre um investigador que possui conhecimentos aprofundados de um certo domínio que oferece contexto científico ao curador de forma a criar uma ontologia orientada para as necessidades do investigador.

O objectivo do nosso trabalho é desenvolver uma ferramenta para ser utilizada no processo de criação de uma ontologia, ajudando os curadores com a extração e visualização de conceitos chave nos documentos de investigação do grupo.

O primeiro passo consistiu numa avaliação de várias técnicas utilizadas na área de aprendizagem de ontologias escolhendo as que se revelaram mais eficientes. As abordagens selecionadas, foram então introduzidas numa plataforma de testes no Dendro. A abordagem resultante é composta por uma ferramenta de Processamento de Linguagens Naturais, diferentes métodos de extração de termos e integração da DBpedia e LOV, de forma a fornecer conceitos e descritores para os termos previamente extraídos.

De forma a poder avaliar as abordagens propostas, fizemos uma avaliação automática e uma manual. Para a avaliação automática foram selecionados 3 casos de estudo, nomeadamente 3 ontologias criadas no Dendro, incluindo o material utilizado na sua construção. A avaliação manual

foi feita com o auxílio dos curadores que construiram essas ontologias. Eles receberam um conjunto de documentos cientificos para usar como input da ferramenta e foram guiados através das diferentes fases, fornecendo feedback em conformidade.

Nós desenvolvemos uma aplicação que foi capaz de extrair a maioria dos conceitos esperados. A opinião dos curadores que avaliaram a ferramenta também foi favorável, sendo que eles concordam que uma ferramenta como o Dendro Keywords é um bom acréscimo para o seu trabalho.

Acknowledgements

I want to start by thanking both of my supervisors, Carla Teixeira Lopes and João Rocha Silva for their continuous guidance and for always being available when necessary. I also want to thank João Castro and Cristiana Landeira for their help during the evaluation of the tool.

I want to thank my family and especially my parents, for all the love and support throughout the different stages of this journey.

Last but not least, I want to thank my closest friends for every moment we have been through and for always being present when I needed.

Cláudio Luís de Sousa Monteiro

"Now I could let these dream killers kill my self esteem, or use my arrogance as the steam to power my dreams"

Kanye West

Contents

List of Figures

List of Tables

Abbreviations

AKE	Automatic Keyword Extraction
ATR	Automatic Term Recognition
CAVA	Communication Audio-Visual Archive
CRCTOL	Concept Relation Concept Tuple based Ontology Learning
DBSCAN	Density-based spatial clustering of applications with noise
DCC	Digital Curation Centre
DCMI	Dublin Core Metadata Initiative
DDI	Data Documentation Initiative
EPSRC	Engineering and Physical Sciences Research Council
FCA	Formal Concept Analysis
GAAC	Group-average agglomerative clustering
HCA	Agglomerative Hierarchical Clustering
ICPSR	Inter-University Consortium for Political and Social Research
IDF	Inverse Document Frequency
JWNL	WordNet Java Library
LOD	Linked Open Data
LOM	Learning Object Metadata
LOV	Linked Open Vocabularies
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
MARC	Machine Readable Cataloging
METS	Metadata Encoding and Transmission Standard
MODS	Metadata Object Description Schema
NCBI	National Center for Biotechnology Information
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NSF	National Science Foundation
OWL	Web Ontology Language
PMI	Pointwise Mutual Information
POM	Probabilistic Ontology Model
PP	Prepositional phrases
RDM	Research Data Management
RAKE	Rapid Automatic Keyword Extraction
RDF	Resource Description Framework
SMES	Saar-bruecken Message Extraction System
SVD	Singular Value Decomposition
SVM	Support Vector Machines
TF-IDF	Term Frequency Inverse Document Frequency
YAKE!	Yet Another Keyword Extraction!

Chapter 1

Introduction

1.1 Context

During the last decades there has been an increase of data production thanks to technological advances. With this increase there has also been a growing concern related to the preservation of data which has resulted in new approaches oriented to solving this situation. A type of data that currently raises big concerns is research data as there are many entities involved, from researchers to institutions and governments, investing money and time on it.

To allow the sharing and preservation of research data, a new area, named Research Data Management (RDM), has surfaced. This area consists of the manipulation of data during research and provides certain benefits such as reducing the risk of loss and increasing efficiency [Ing16]. However this is still a complex problem among researchers. They often argue that managing data is very complicated and that they should rather focus on their work [RRL12].

One of the RDM sub domains is data description, which should start as early as possible. For this, metadata schemas are used, but since there are a wide variety of different domains, these schemas might prove to be inadequate [CPA+15].

Dendro was created having these problems into consideration. It provides a collaborative and user-friendly interface in which researchers might chose the set of descriptors that fit their needs from existing ontologies [RRL18].

Ontology Learning is a relatively new area with the aim of decreasing the costs associated with the creation of an ontology [CASJ09]. It benefits from the advances in areas such as data mining and informational retrieval and is the process of identifying terms and concepts with the aim of creating an Ontology [WLB12].

1.2 Motivation and goals

Currently, the process of creating an ontology is both time and resource consuming. Also, many organizations do not possess the knowledge to create them.

In order to create an ontology there must be a meeting between a researcher that is usually an expert in a certain domain and a curator who provides certain knowledge regarding ontologies and RDM. After this, the curator proposes a list of concepts regarding that domain which will have to be validated by the expert [SRC12]. With these aspects in mind there is a great interest in automating the process.

The main objective of this work is to implement a tool that assists curators in the task of creating an ontology by the team that is working on Dendro.

In order to do this, different ontology learning techniques, which are based on areas such as information retrieval and machine learning, were explored in order to find the better approach towards this goal. The end result is able to provide the curator a visual representation of the domain based on a collection of scientific documents. This is done by presenting a list of the most important terms in the document and if available an associated concept and a descriptor.

1.3 Contributions

Our work was presented as a Flash Talk in the 3rd Fórum GDI[1] which occurred in November, 2017 [fora]. It was also accepted for publication and will be present as a poster in TPDL2018[2] which will happen in September, 2018 [MLS18].

1.4 Dissertation Structure

In addition to the current chapter, this document contains 6 more chapters. Chapters 2 and 3 are related to the state of the art, specifically the areas of research data management and ontology learning. In Chapter 4 we make an evaluation on existing keyword extraction techniques in order to find the most suitable to implement in our tool. Chapter 5 discusses the implementation of the tool. We provide an overview between the different phases, including the relations between them. In Chapter 6 we evaluate the tool previously described using both an automatic and a manual evaluation. The automatic evaluation was done resorting to gold standard ontologies while the manual evaluation was done by the curators that built those ontologies. Chapter 7 provides a summary of the whole process including the work to be done in the future.

[1] http://forumgdi.rcaap.pt/3forum/
[2] http://www.tpdl.eu/tpdl2018/

Chapter 2

Research Data Management

In this chapter we introduce Research Data Management and its importance for research. We then provide a definition on metadata and why it is important for managing data followed by different types of schemas. In the end we provide an overview of different data managements platforms.

2.1 Introduction

The importance of research data management is widely recognized and there is a general agreement that it should start as early as possible in the research workflow to minimize the risk of data loss. The adoption of proper data management practices provides advantages for everyone involved, from researchers to research institutions. Surkis and Read [SR15] state that *"Data management ensures that the story of a researcher's data collection process is organized, understandable, and transparent."*. A good example of the importance of data management is the need of a researcher to use work that was previously done by others; in this case, handling a wide variety of raw data without proper management might be too difficult if not impossible.

2.2 Research Data

Different definitions exist for research data, with one of the reasons being the variety of domains available. However, there is one definition provided by the Engineering and Physical Sciences Research Council (EPSRC) that is widely used and defines research data as *"recorded factual material commonly retained by and accepted in the scientific community as necessary to validate research findings"* [Bur12]. Research data can also be divided into categories such as Observational or Experimental data and, depending on the research area, might be of a various formats such as spreadsheets, photographs, schema, among others [Wha].

2.3 Data sharing and preservation

With the recent increase in the production of research data, particularly in the last decades, there has been a growing concern within the scientific community about the consequences of the massive production of data. This is mainly due to the increase of research activity and the growing volumes of data that are generated because of the improvements in research methods and tools [RSRF10]. Fields, such as Astronomy and Climatology— where work is strongly supported by data analysis— have already been relying on the sharing of data between research for some time and possess well implemented norms and infrastructures [RSRF10]. Data sharing is not as easy as it seems though, and there are many reasons why researchers do not do it [Bor12]. Examples include lack of knowledge or incentives, or even because the data is not in a transferable form.

Borgman [Bor12] states several reasons why data should be shared:

- **Reproduce research** - The reproduction of data is the most important argument for data sharing. There is a need for data to be shared since peers must be able to evaluate the validity and reliability of the research.
- **Make results of publicly funded research available to the public** - This is the simplest of the reasons provided, and means that data should be public if the resources used to fund the research came from taxpayers. It has already proved to be successful in some cases like the biomedical community [Bor10].
- **Enable others to ask new questions of existing data** - Sharing data allows others to pick up existing data by either using a set from a single individual or combining multiple ones and start *"asking questions"*, which in the end, is one of the bases for science.
- **Advance the state of research and innovation** - Fields that heavily rely on data such as Astronomy or specific areas of Biology are much more open to the idea of sharing data, improving more than fields that are not inclined to share [Bor10].

2.3.1 Data Citation

We have shown how important the sharing of data is to the advancements of different areas. However, since most of the work is done by the researcher, the researcher should feel motivated to describe and share his data.

Data Citation is not only used to identify the data used but it is also a way to recognize the authors [PDM10]. The increase in data citation shows that the person cited provided work that is considered important and can be used by others that are willing to build on it [DJ86]. Data citations can also be used in research funding and determining salaries since it is an indicator of the quality of the data.

Piwowar et al. [PDF07] provided a study, in which they observe the relation between the citation count and if the data was made open, and they found that cancer clinical trials which shared their data were cited about 70 % more than the ones who did not.

2.4 Digital Curation

Digital curation offers benefits in both the short and long-term [Abb08]; some short-term benefits include the improvement of the quality and trustworthiness of data , and can benefit from the adoption of common standards which help with collaboration. Long-term benefits include the protection against data loss and obsolescence, encouragement of data re-use and the use of tools and services to allow the migration of aspects such as metadata. Digital curation also has its problems, it can be costly and require substantial amounts of time and resources, issues that can be problematic, especially in small groups [Abb08]. Since Digital Curation responsibilities can be shared between different institutions and that different fields use terminology in different ways, it may lead to either misunderstandings or inconsistencies. The Digital Curation Centre (DCC) has developed the Curation Lifecycle Model [Hig08] that can be seen in Figure 2.1. This model provides an overview of the necessary stages for proper data curation and it can also be used to plan activities within an institution. At the center is data, which can be in any form. On top of it are the lifecycle actions, the first being *Description and Representation Information*, which is where this work is focused and consists on assigning metadata using the correct standards to ensure the control on the long term.

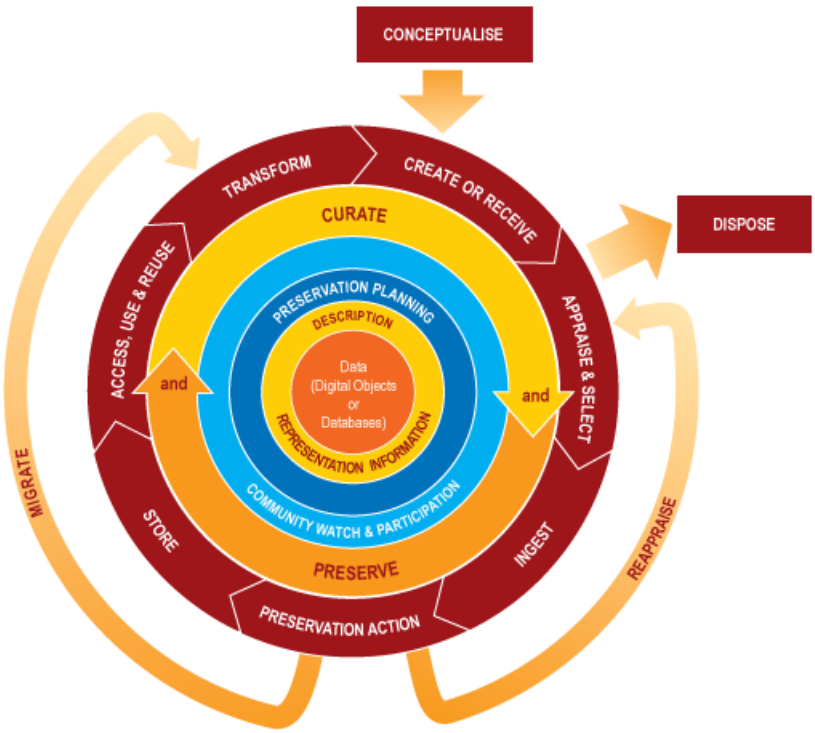


Figure 2.1: Key elements of the DCC Curation Model [Hig08]

The University of Porto[1] in conjunction with the University of Minho[2] carried out a study in which they provide five different ways to treat data curation [RSRF10]:

Curation by scientists and experts that use the data— In this case, there is no set of rules regarding curation and it depends mostly on the people involved. It can be found in Universities or Research centers. CAVA[3] (Communication Audio-Visual Archive) by the University College London[4] is an example of this case.

Curation by scientific organizations and institutions— Usually happens when a necessity to preserve a set of data arises or even offering services within a community or research domain. It usually happens among Universities and has the strong point of reuniting researchers which are interested in that subject. An example is the NCBI[5] (National Center for Biotechnology Information) in the USA.

Curation by universities and research centers— Is similar to the previous one, although in this case the initiative comes from a University or research center and because of this tends to include a variety of domains. The two most used forms of curation integration are computation centers and libraries. Datashare[6] from the University of Edinburgh[7] is an example of this case.

Curation by official organizations— It comes from the organizations that administrate science at a country level. They usually have a set of rules and support infrastructures, which makes this a strong point. However, the distance between the researchers and the service and the necessity of hosting a variety of data from different fields, might prove to be a problem. DataONE[8] (Data Observation Network for Earth), financed by the NSF[9] (National Science Foundation, USA) is one example.

Curation by informal communities— The existence of communities that hold both specialists and amateurs will probably be a common sight in a few years. Because of how the communities are set, it is probable that there will be multiple replicas of the same dataset which will prevent their disappearance; however, it might be hard to identify its creator. Wikispecies[10] is an example of this type of digital curation process.

2.5 Metadata

At the center of data management is metadata, with a variety of standards defined, Data curators can choose generic metadata models that will suit a wide variety of data, or if needed, may choose others more oriented towards a specific domain. Metadata is usually defined simply as *data about data*. Later, Jane Greenberg [Jan88] stated that this definition is too ambiguous given the different

[1] http://up.pt
[2] https://www.uminho.pt/
[3] http://www.ucl.ac.uk/ls/cava/
[4] https://www.ucl.ac.uk/
[5] https://www.ncbi.nlm.nih.gov/
[6] https://datashare.is.ed.ac.uk/
[7] https://www.ed.ac.uk/
[8] https://www.dataone.org/
[9] https://www.nsf.gov/
[10] https://species.wikimedia.org/

use of the term and defined metadata as *"structured data about an object that supports functions associated with the designated object."*

There are a few key aspects about metadata that make it important for research data management [Bac08]. The first is that metadata allows the discovery of relevant information more easily by either helping to find resources or bringing similar resources together, helping with object differentiation. Another aspect is that, if properly represented, it may also increase interoperability, which means that systems will be able to exchange information between each other without much concern for information loss. Good metadata will also allow the preservation of the work, which will allow it to be available in the future.

2.5.1 Types of Metadata

According to the National Information Standards Organization there are three main metadata categories [Pre04]:

Descriptive metadata consists of attributes on the object being described, for purposes such as discovery and identification. Examples of this type are elements such as title, author or keywords.

Administrative includes the information that helps managing a resource, such as the file type or who is allowed to access it. Administrative metadata can be divided in three subcategories: Technical metadata which describes the necessary information to access the data and examples are file size or file type; Preservation metadata refers to the elements related to the preservation management of an information resource; Rights metadata concerns the intellectual property rights related to the content, such as copyright status or license terms.

Structural provides details about the internal structure of the resource. It is an important indicator of its context. An example is the table of contents.

2.5.2 Metadata Schemas

Greenberg [Jan88] defines Metada Schema as *"A unified and structured set of rules developed for object documentation and functional activities."* With this aspect in mind, several schemas have been defined and can be of two types. Generic can be used for elements such as the author name and the title of a book and can be easily understood by people without much knowledge and the other one is domain oriented and is supposed to be used by experts depending on their needs.

2.5.2.1 Generic Metadata Schemas

These schemas provide descriptors that fit a variety of different systems and needs. They can be used by most individuals, seeing that it does not require much knowledge of any specific research domain and allow for better interoperability, but can only be used for general aspects. Some of the most well known schemas are Dublin Core, MARC, MODS and METS.

Dublin Core was first proposed in 1995 by the DCMI (Dublin Core Metadata Initiative) [ANS01], which began with a workshop that brought together individuals such as librarians and content specialists in order to help improve standards for information discovery. The basic idea

was to create a model that was simple enough to be used by anyone instead of only by data management professionals. The earlier versions of the model contained only 13 elements but, in 1998, a new version with 15 elements, which can be seen in Table 2.1, was published [Dub03]. These elements are widely understandable and can be used in a wide range of areas. Due to its simplicity, its popularity has been increasing steadily with the standard being adopted by CEN/ISSS[11] or even being endorsed by government from various countries for promoting the discovery of information in digital form.

Content	**Intellectual Property**	**Instantiation**
Title	Creator	Date
Subject	Publisher	Type
Description	Contributor	Format
Source	Rights*	Identifier
Language		
Relation		
Coverage*		

Table 2.1: The 15 Dublin Core elements. Elements with (*) are the ones introduced in the newer version [Dub03]

.

MARC stands for Machine Readable Cataloging and was developed during the 1960s by the Library of Congress. It is the most used metadata language by librarians and it was created because computers needed a mean to interpret information from a catalog card [Pre04]. Since books have different title sizes and fields, the model needs to be flexible, so MARC allows records with an unlimited number of entries and unlimited length [Fur00]. Using MARC as the standard reduces redundant cataloging work and promotes sharing of information between libraries. The standard MARC later evolved to MARC21[12] which is the primary source of authority records in the US and most of the English-speaking countries and is currently maintained by the Library of Congress [Fur00]. Each record is divided into fields such as author or title and can then be divided into subfields, each field is then associated with tags which consists of a 3-digit-number [Fur00]. An example of a field with subfields can be seen on Figure 2.2.

```
300 ##   $a 675 p. : $b ill. ; $c 24 cm.
```

Figure 2.2: Example of a MARC entry. The first number represents the physical description, $a the number of pages, $b illustration information and $c the dimension [Fur00]

.

[11] https://www.cen.eu/Pages/default.aspx
[12] https://www.loc.gov/marc/bibliographic/

MODS (Metadata Object Description Schema) is a metadata standard that was developed by the Library of Congress and MARC Standards Office. MODS is written in XML, which makes it more robust and software independent. It is based on a subset of MARC, but unlike the latter, instead of using numbers as field names, it uses English words which does not force the user to possess knowledge about MARC [Gar03]. When compared to Dublin Core, MODS offer a more complete and specific set of elements, offering 20 top-level elements, which can be seen in Table 2.2. MODS possesses features that allows the addition of different metadata to a record [Pre04]. An example of a description using MODS can be seen in Figure 2.3.

titleInfo	name	typeOfResource	genre
originInfo	language	physicalDescription	abstract
tableOfContents	targetAudience	note	subject
classification	relatedItem	identifier	location
accessCondition	part	extension	recordInfo

Table 2.2: The 20 MODS elements as seen in [Pre04]

.

```
<titleInfo>
    <title>Sound and fury : the making of the punditocracy / </title>
</titleInfo>
<name type="personal">
    <namePart>Alterman,Eric.</namePart>
    <role>creator</role>
</name>
<typeOfResource>text</typeOfResource>
<genre authority="marc">bibliography</genre>
<publicationInfo>
    <placeCode authority="marc">nyu</placeCode>
    <place>Ithaca,N.Y.:</place>
    <publisher>Cornell University Press,</publisher>
    <dateIssued>c1999.</dateIssued>
    <dateIssued encoding="marc">1999</dateIssued>
    <issuance>monographic</issuance>
</publicationInfo>
```

Figure 2.3: A description using MODS [Ben05]

.

METS (Metadata Encoding and Transmission Standard) is a data communication standard that, like MODS, is expressed using XML and is an initiative of the Digital Library Federation[13] and it is currently maintained by the Network Development and MARC Standards Office of the Library of Congress [Can05]. METS was developed for the scalability, interoperability and the preservation of digital library objects. It can be divided in seven different sections [Can05]: *METS Header, Descriptive Metadata Section, Administrative, File Group Section, Structural Map, Structural Map Linking and Behavior Section.* Although it contains seven sections, the only one that is mandatory is the *Structural Map*, which is refereed as the core of a METS model [Can05], and its

[13] https://www.diglib.org/

purpose is to outline the hierarchical structure of an object. An example of a Structure Map from a METS document can be seen in Figure 2.4.

```
<mets:structMap>
    <mets:div TYPE="compactDiscObject">
        <mets:div ORDER="01" TYPE="Track"/>
        <mets:div ORDER="02" TYPE="Track"/>
    </mets:div>
</mets:structMap>
```

Figure 2.4: Structure Map from a METS document [Cun04]

2.5.2.2 Domain-specific Metadata Schemas

Generic models allow the description of resources in a very general way. So, when the necessity to describe more specific scientific domains arises, more specific models should be used. This sort of schemas are used when generic ones are not sufficient, namely when recording the data production context of datasets from specific research domains. Since these schemas are more specific, they are usually only handled by experts. Some examples include certain domains such as Biology (Darwin Core), Engineering (LOM) and Humanities and Social Sciences (DDI).

Darwin Core first appeared around late 1990s and is a standard for sharing data about biodiversity [WBG+12]. It resulted from the necessity to provide proper data about species and their relation to the environment. Its main purpose is to provide a common language for sharing biodiversity that reuses standards from other domains whenever it is possible. Darwin Core can be seen as a extension to Dublin Core for biodiversity [Dar], and has the goal of making data sharing easier while improving their reuse since Darwin Core uses a well-defined standard. Since the model may not always fit the user needs it can also be extended allowing the insertion of new terms. A list of the nine different terms that constitute the model, followed by examples, can be seen in Figure 2.5.

Record-level Terms	Dublin Core terms, institutions, collections, nature of data record	Simple Darwin Core (flat)
Occurrence	evidence of species in nature, observers, behavior, associated media, references.	
Event	sampling protocols and methods, date, time, field notes	
Location	geography, locality descriptions, spatial data	
Identification	linkage between Taxon and Occurrence	
Taxon	scientific names, vernacular names, names usages, taxon concepts, and the relationships between them	
GeologicalContext	geologic time, chrono-stratigraphy, biostratigraphy, lithostratigraphy	
ResourceRelationship	explicit relationships between identified resources (e.g., one organism to another, taxon to location, etc.)	Generic Darwin Core (relational)
MeasurementOrFact	measurements, facts, characteristics, assertions, references	

Figure 2.5: A table consisting of the nine Darwin Core categories [WBG+12]

.

LOM (Learning Object Metadata) was published by the Institute of Electrical and Electronics Engineers Standards Association[14] and is a standard for the description of learning objects which is defined by the authors as *"any entity, digital or non-digital, that may be used for learning, education or training"* [Bar05]. LOM consists of a hierarchy of nine categories, each with its own subcategories. A visual representation of the LOM model can be seen in Figure 2.6.

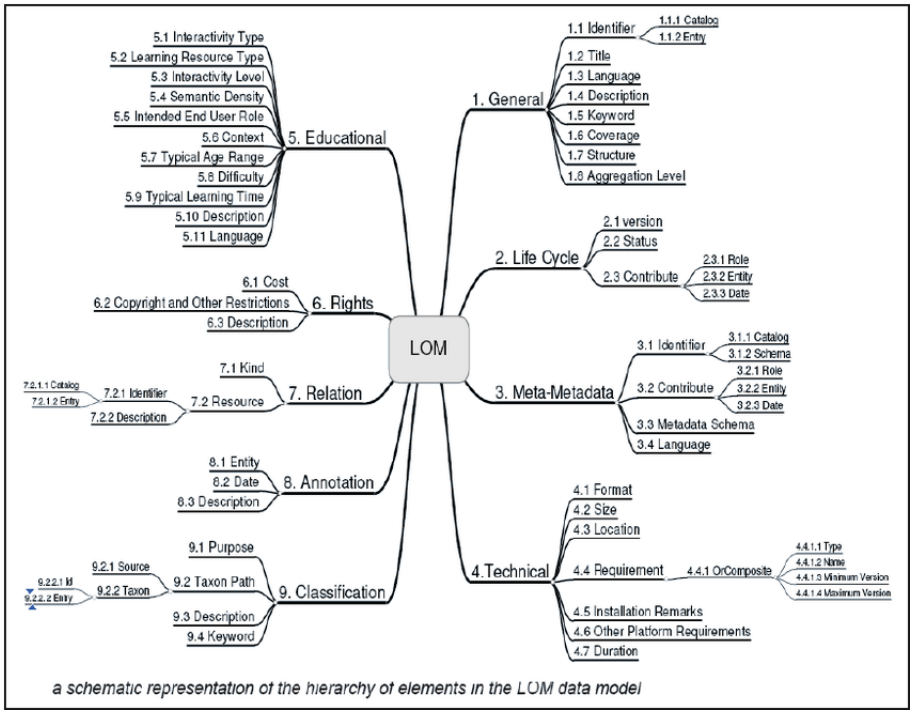


Figure 2.6: A representation of the hierarchy in the LOM Model [Bar05]

DDI (Data Documentation Initiative) was initiated in 1994 by the Inter-University Consortium for Political and Social Research[15] (ICPSR) and its original goal was to replace the widely used OSIRIS model with one that was more modern [Rys02]. It has now become a key standard in areas such as social and economic sciences [Pre04]. DDI is usually defined as a number of XML schemas which are separated into modules. The elements are arranged in a tree-like structure and consists of: *The Document description, The Study description, The Data Files Description, The Variable Description and Other Study-Related Materials* [Rys02]. An example of a Document Description can be seen in Figure 2.7.

[14] http://standards.ieee.org/
[15] https://www.icpsr.umich.edu/

```
<docDscr>
    <citation>
        <titlStmt>
            <titl>United States Historical Election Returns, 1824-1968</titl>
            <IDNo>1</IDNo>
        </titlStmt>
        <prodStmt>
            <producer abbr="ICPSR">
                <ExtLink URI="http://www.icpsr.umich.edu/images/icpsr-logo.gif"title="ICPSR Logo" role="image"/>
                ExtLink URI="http://www.icpsr.umich.edu/"title="URL of ICPSR Web Site"/>
            </producer>
        </prodStmt>
        <verStmt>
            <version date="2010-09-09"></version>
        </verStmt>
        <holdings URI="/cocoon/xml/STUDY/00001.xml"></holdings>
    </citation>
</docDscr>
```

Figure 2.7: Example of a Document Description from a DDI document

2.5.3 Application Profiles

There is a wide range of research domains, which means that most likely their needs for metadata are not the same. This often means that the group will have problems in finding a standard that will fit their needs [Ric44]. Since there is not a single metadata model that can fulfill the requirements of every application, it became important to be able to cross domains. As an answer to this, *Application Profiles* were introduced, they provide the creators a mean to use a *mix-and-match* approach and can be defined as "a mixture of metadata elements selected from one or more schemas and combined in a compound schema" [DHSW02]. They possess a few characteristics [RH00]: They are able to use elements from one or more sets, but they cannot create elements that were not previously in these sets; If Creators wants to introduce an element that does not exists elsewhere, the Creator must first create his own schema; It can refine the definition of an element, but only if it is to make it more specific; It may define relations between elements and their values, such as a specific element may require the presence of another or a value of a element depends on the value of another.

2.5.4 Ontologies

An ontology can be defined as a *"formal, explicit specification of a shared conceptualization, which can be used to provide a shared and common understanding of a given domain"* and its role is to help the construction of a model for a domain by providing certain terms and relations [SBF98].

Ontologies can be of two types [GPR10]: *Lightweight* include concept hierarchies and taxonomies and *Heavyweight* which add axioms to the first type, thus making them more complete but also more complex.

For the representation of Ontologies, standards such as Resource Description Framework (RDF) and Web Ontology Language (OWL) can be used as they provide certain benefits such as: the ability to put together information from a variety of sources, support for semi-structured data, syntax separation from syntax modeling, web embedding and resilience to change [RTMC05].

2.5.5 Linked Open Data

According to Berners-Lee et al. [BHBL09] Linked Data is about creating links among data from different sources by using the Web, with Linked Open Data (LOD) being linked data that is open content.

For data to be considered *open* it should follow some principles: all public data must be made available and should be accessible by a variety of users without the need for registration, it should be collected from the source, should be available as fast as possible for preservation and must be free of control and free of licenses [Ope].

LOD plays an important role regarding these principles. It is becoming very important for data management due to its ability to use different domains unlike conventional platforms [BK11]. Since it is focused on the quality of metadata, LOD is crucial to the increase of the value of data and improving its availability.

Linked Open Data has already been applied to a variety of areas. Examples are DBpedia[16] and The OBO Foundation[17].

RDF triple store is used to store semantic data [Tri]. This data is stored in a way that every element has links between them, which in the end will allow for more flexibility and lower costs. Ontologies are supported since they allow the specification of classes and their relations. Data in RDF is stored as "subject->predicate->object" with the predicate showing what the relation between the subject and the object is.

Triple stores are currently being used to represent LOD, allowing the connection between different datasets, and making the search in different origins less costly and quicker [Tri].

2.6 Data Repositories

At the moment several research teams have adopted known platforms to manage and share their research, having several individuals that range from researchers to data curators involved in the description of data. This data plays an important role in defining the requirements for a data repository. Due to its increase in popularity, RDM platforms tend to implement features that will facilitate the work of the people involved. In Figure 2.8 we can see both the support for triple store and a comparison between each platform regarding the metadata requirements which was extracted from the work of Amorim et al. [ACdSR15].

2.6.1 DSpace

DSpace is a free and fully customizable repository of publications and bibliographic information that may be adapted for data [DSp]. It allows the customization of aspects such as the user interface which will help the integration of the platform with other resources by the institution. Regarding metadata, although Dublin Core is the default format, DSpace allows the use of other hierarchical

[16] http://wiki.dbpedia.org/
[17] http://obofoundry.org/

metadata schemas such as MARC or MODS, but this may require external tools. DSpace provides Authentication mechanisms as plugins, but if needed institutions may build their own authentication protocol. It offers compatibility with the following standards Standards compatiblility by including support to standards such as OAI-PMH[18], OAI-ORE[19] or SWORD[20]. It allows the user to configure a PostgreSQL or Oracle database where DSpace manages its metadata.

DSpace is currently used by certain organizations such as government and educational institutions. An example is the University of Porto with both Repositório Aberto[21] and Repositório Temático[22].

2.6.2 CKAN

Unlike DSpace which has its roots in publications management, CKAN is a fully-featured open source RDM platform [CKA]. It allows the organization to either chose from a set of community developed extensions or create their own. As for metadata, it only requires minimal additions. It is currently used by a variety of institutions such as Data.gov[23] or the European Data Portal[24].

2.6.3 Figshare

Figshare is an RDM platform oriented towards academic institutions. It allows their users to store and share their research outputs [Fig]. Unlike the previous platforms where the only costs are maintenance related, Figshare charges a monthly fee. Figshare allows the integration with other repositories such as DSpace, and allows the addition of custom metadata. Examples of users are The University of Auckland[25] and The University of Melbourne[26].

2.6.4 Zenodo

Zenodo is an Open Source RDM platform which was created by OpenAIRE[27] and CERN[28] [Zen]. It provides flexible licensing and allows the creation of custom communities. It associates a DOI to each upload, that will make it more easily accessible. As for metadata it allows it to be exported as MARCXML and Dublin Core.

[18] https://www.openarchives.org/pmh/
[19] https://www.openarchives.org/ore/
[20] http://swordapp.org/
[21] https://repositorio-aberto.up.pt/
[22] https://repositorio-tematico.up.pt/?locale=pt
[23] https://www.data.gov/
[24] https://www.europeandataportal.eu/
[25] https://www.auckland.ac.nz/
[26] http://www.unimelb.edu.au/
[27] https://www.openaire.eu/
[28] https://home.cern/

Category	Feature	Dspace	CKAN	Figshare	Zenodo
Metadata	Rquired Fields	Title, Date of issue	Title	Author, title, categories, description	Type, DOI, author, title, description
	Exporting Schemas	Any pre-loaded schema	No	DC	DC, MARCXML
	Schema Flexibility	Flexible	Flexible	Fixed	Fixed
	Validation	Yes	No	No	Yes
	Versioning	No	Yes	No	No
Triple Store	Support	Yes	Yes	Yes	No

Figure 2.8: Comparision between platforms regarding metadata and triple store features. Metadata features extracted from the work of Amorim et al. [ACdSR15]

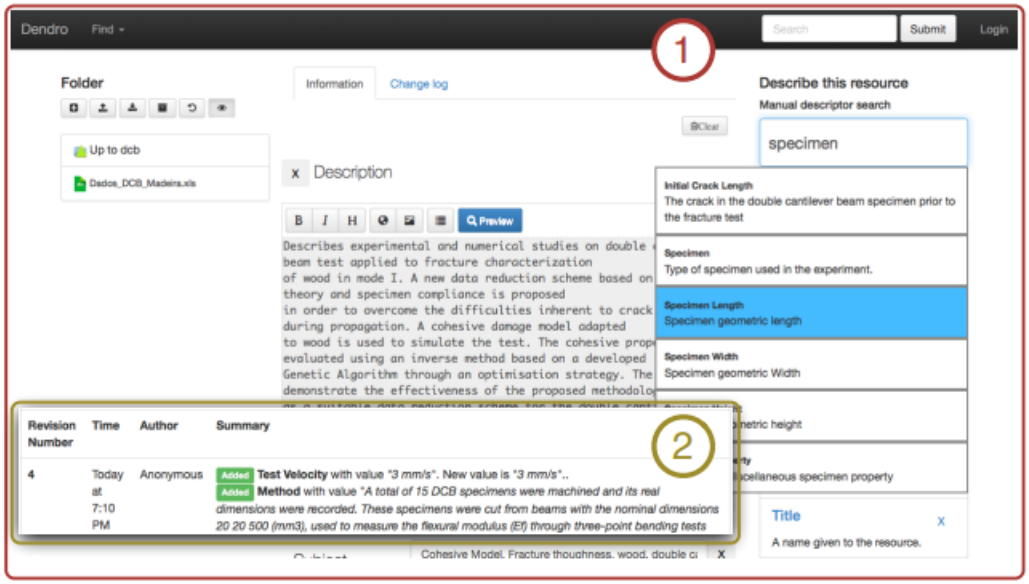
2.7 Dendro

Dendro is a research data management platform that focuses on the early stages of research due to their importance. It was developed as a collaborative system for users to describe their data, allowing them to chose their own metadata standards from existing ontologies [RdSACRCL14]. In Figure 2.9 we can see an overview of the interface, Dendro provides a search box where the user can look for data descriptors and since it offers a collaborative environment, a record of these changes are kept .

Dendro applies a graph-based model which offers certain advantages in comparison to relational models [dSRL14], such as the improvement of the system flexibility and when searching for anything, knowledge of the domain is more relevant than the knowledge of metadata schemas. The fact that the model is based on ontologies allows the integration of LOD functionalities more easily.

Dendro makes use of a triple store which provides a relation between the key and the value as explained in Section 2.5.5 and allows the connection between different resources [dSCRL14].

An example of a file description in Dendro can be seen in Figures 2.11, 2.11 and 2.12, starting on the project home page, followed by the selection of a file that currently does not have a description and the addition of certain terms from Dublin Core. Following this example, we can see that data description in Dendro is done manually by the researcher, which is what this work aims to improve, through partial automation of the process. This also shows how important is this work since having to describe data was one of the reasons given by researchers as to why they do not manage their data.



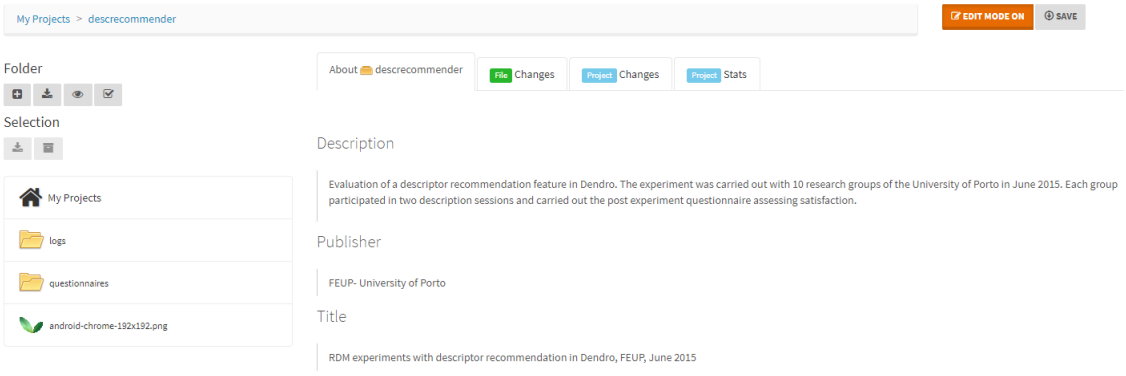


Figure 2.10: Home page of a project in Dendro

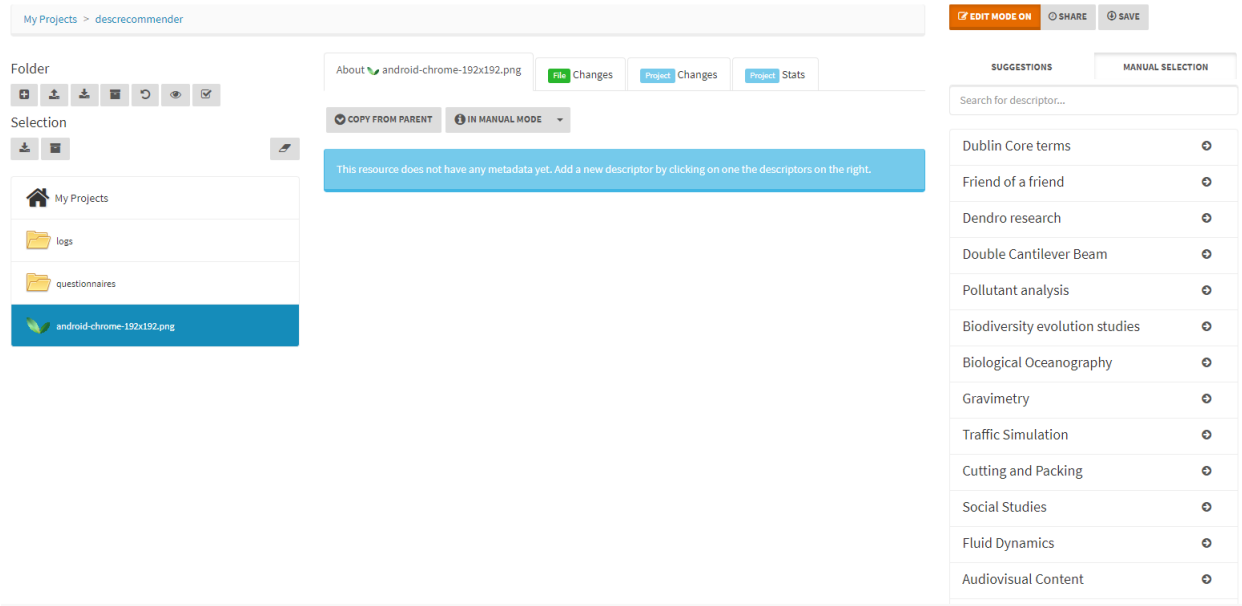


Figure 2.11: Selection of a file without a description

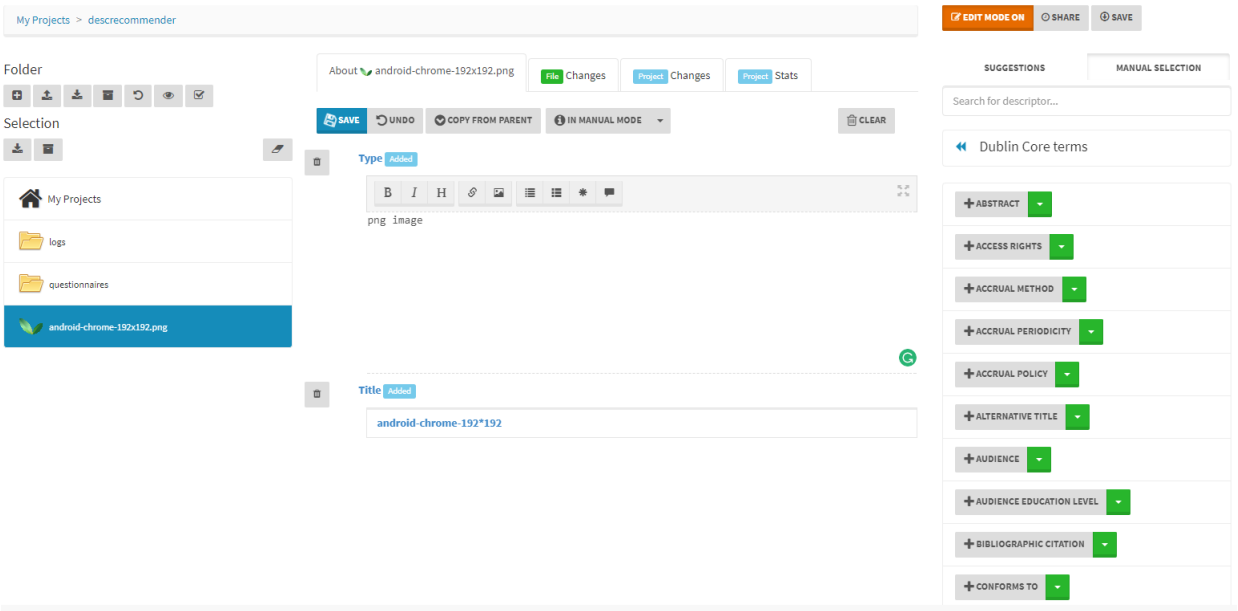


Figure 2.12: Addition of Dublin Core terms Type and Title to a file

2.8 Summary

This chapter allowed us to see the importance and the necessity of managing research data. It was also described how important metadata is for data management including keys aspects that support this. Also different metadata schemas that are based on certain needs were presented. The concept of Linked Open Data was introduced since it is becoming very important for data management. In the end a few data management repositories were described. These platforms

offer help to everyone involved in the process of managing data by allowing the implementation of certain features based on their needs.

Chapter 3

Ontology Learning

In the previous chapter we gave an introduction on what is an ontology. In this chapter we will introduce the Ontology Learning area followed by existing techniques and learning systems.

3.1 Introduction

Currently the task of engineering an ontology is still an expensive and resource-demanding task [CASJ09]. This proves the need to implement methods and techniques aimed at decreasing the costs associated with creating and maintaining an ontology. Advances in areas such as information retrieval, data mining and machine learning have shown to be fundamental and have provided means to extract and improve the management of information [WLB12]. Following these advances, a new area of research has been born, known as *Ontology Learning* [WLB12] which is the process of identifying, for example, terms and concepts with the aim of creating or maintaining ontologies. This not only reduces costs [HRR11], but also assists in creating an ontology that better fits the needs of the user. Ontology Learning is still considered to be a semi-automatic process because, although there have been improvements in the areas mentioned above, there is still the need to have human intervention [CASJ09].

3.1.1 Ontology Learning Layer Cake

Ontology Learning is composed of various tasks, which have been aggregated in a layer diagram named *Ontology Learning Layer Cake* [CASJ09] that can be seen in Figure 3.1. This approach was implemented in a way that the output of a layer works as an input for the upper layer. In his work, Wong [Won09] provides an overview on the types of outputs in this model: *Terms* are at the base of the layer model and usually consists of simple words; *Concepts* can be abstract or concrete and are formed by grouping similar terms together; *Taxonomic relations* consists of the hierarchy between concepts; *Non-taxonomic relations* are the interactions between concepts; *Axioms or Rules* are propositions that are always taken as truthful and work as a base for deducting other axioms;

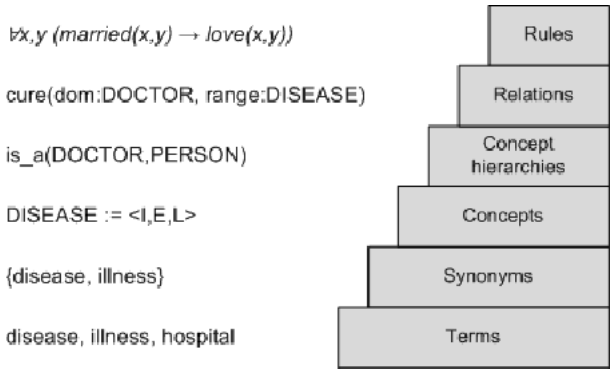


Figure 3.1: Ontology Learning Layer Cake [BCM05]

3.1.2 Ontology Learning Tasks

Cimiano et al. [CASJ09] provided an overview of the various tasks from each layer:

Term Extraction is one of the tasks where the objective is to extract and arrange terms into groups of synonyms. A simple example of this step is counting the frequency at which a term appears in a set of documents. However, advances in the area of informational retrieval have provided much more efficient methods such as tf–idf [CASJ09]. Certain domains will also require adaptations of these models to fit their traits.

During **Synonym Extraction** the goal is to find synonyms. Approaches for this task are mostly based on *Distributional hypothesis* which states that words tend to have the same meaning if they are used in a similar context [Sah05]. For this purpose, Dekang Lin [Lin98] proposed a method that consists of dividing the amount of characteristics in common between two objects by the quantity of information in their description. This method uses triples that represents two words and the relation between each. The characteristics in common are given by the triples that appear in the description of both. This description consists of the frequency count of every triple that starts with the word. The amount of information in a description is given by the sum of every frequency count. The lexical database *WordNet* is widely used in this context [Mil95].

Concept Learning is the task that consists of searching and classifying information that allows the differentiation between objects. In this section Cimiano et al. [CASJ09] distinguish three paradigms named Conceptual Clustering, Linguistic Analysis and Inductive Methods. In *Conceptual Clustering* a classification scheme is formed based on descriptions from the input objects and consists of a set of clusters that may or may not be ordered hierarchically [JHC01]. It usually adds a description to a cluster based on attributes that it shares with others. An example of a conceptual clustering approach is COBWEB [Fis87] which executes hill-climbing through a classification scheme using operators that facilitate bidirectional movement and adds objects to the classification tree in an incremental form in which each node represents a cluster. The other two are *Linguistic Analysis* and *Inductive Methods*, the first may be applied to extract a description of a concept as a Natural Language Description and the second may be applied to derive descriptions from a group of instances.

The **Concept Hierarchy** task provides the taxonomic relations and its objective is to place each concept correctly in the hierarchy. Approaches such as *Clustering* and *Classification* may be used.

The following task consists of finding **Relations**, more specifically, non-taxonomic relations among words. This can be achieved by applying machine learning and natural language techniques, something that can be done by finding similarities between words that are relatively close, usually a sentence. Maedche and Staab [MS00] defined an approach in which they define both the relations between words and the level of abstraction at which relations should be determined.

The last task of ontology learning is **Axioms and Rules** and in this phase axioms are generated from the previous concepts. Volker et al. [VHC07] provide an implementation in which they provide axioms based on a syntactic analysis of natural languages.

3.1.3 Learning from Structured Data

Structured data usually consists of databases or existing ontologies. Since databases make up for a considerable part of available structured data and are considered an essential part of information systems [Alf10] they make a good source for ontology learning. There have been a few proposed approaches regarding structured data in which the biggest problem is usually defining the parts of the data that provide the necessary information [DG08]. An approach based on structured data was presented by Jacinto and Antunes [JA12] in which they provided a method for building an ontology based on relational databases by allowing the user to choose from a pre-determined set of rules that allow the conversion of elements such as entities to concepts and relations in the ontology. Another was presented by Li et al. [MXS05] where they utilize a group of rules to automatically learn an ontology.

3.1.4 Learning from Semi-structured Data

Semi-structured data is a mixture between structured and unstructured data. It combines unstructured text with elements such as metadata. Examples of semi-structured Data are BibTeX files because although it looks structured, it can not really be compared to a database as fields may be missing or have features that are hard to describe [SVGK16]. XML is also another example of semi-structured data.

Certain approaches have been proposed for ontology learning from semi-structured data. In their work, Davulcu et al. [DV04] provide a method which converts HTML pages provided by the user to structures such as XML so that they can be mined for generating taxonomies. It uses tree-mining algorithms to determine the key domain concepts and the relations between each. Hazman et al. [HBR09] also developed a method in which they combine two approaches, the first utilizes the HTML headings and the second its hierarchical structure to be able to define concepts and the taxonomical relations between them. Apart from the HTML pages, the method also receives as an input a set of *"seed concepts"* that represent key concepts in the domain. The system builds two

different ontologies, one based on N-grams and another based on the HTML structure In the end merges both while making the necessary adjustments.

3.1.5 Learning from Unstructured Data

Unstructured data make up for most of the data available and can be of certain types such as text documents [DG08]. It is usually the most difficult [HRR11] type of data to learn from the available types and the one that still requires the most research. Natural Language Processing (NLP) is widely used for this type of data and an approach based on such method was provided by Sabou et al. [SWGM05]. The approach takes advantage of the syntactic regularities that are fundamental from the sublanguage nature of the web service. There are three groups of syntactic patterns that are used to collect the different types of information: *domain concepts* are usually characterized by the nouns in the body of the text, then verbs are used to identify *functionalities* from the nouns adjacent to the verbs in question and *relations* between the terms that are interrelated by prepositional phrases (PP). These relations can then be converted to an ontological relation.

3.2 Ontology Learning Techniques

In the previous section we provided a brief overview of the relation between the tasks and the different outputs, here we present the techniques related to each task. Figure 3.2 provides an overview of the whole process, where inside the circles are highlighted the areas to which this work will contribute.

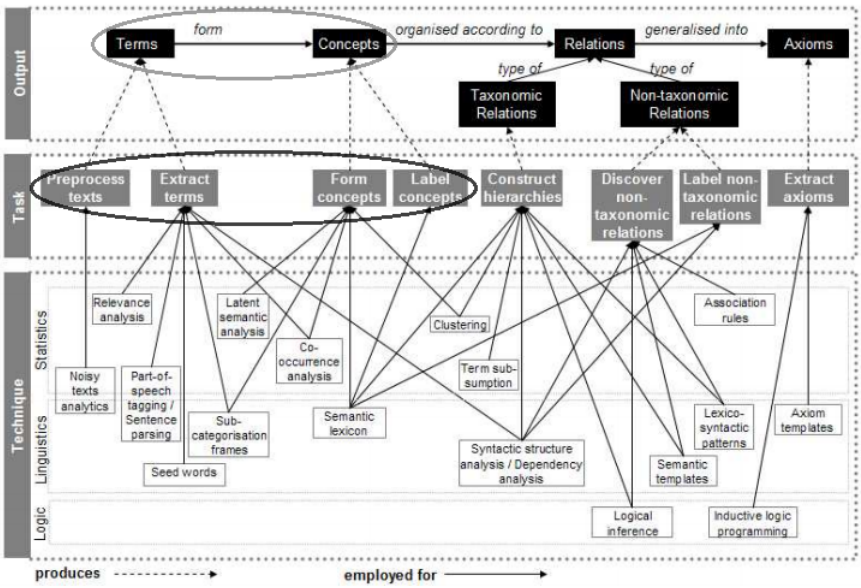


Figure 3.2: The relation between the outputs, tasks and techniques in Ontology Learning. Adapted from the work of Wong [Won09]

3.2.1 Statistics-based Techniques

Techniques based on statistics usually come from the fields of machine learning, information retrieval and data mining and consist of methods that are based on a statistical approach to define new concepts [WLB12].

The ability to retrieve relevant information is the main goal of Information Retrieval and shows the importance of a certain resource towards the needs of a user [Bor03]. Different approaches have been used for **Relevance Analysis**. The Probability Ranking principle was proposed by Robertson [Rob77] in which the author states that references should be ranked by probability according to its utility for the user, and calculates this probability based on all the available data. Term Frequency Inverse Document Frequency (TF-IDF) can also be used for relevance analysis [REE03]; higher TF-IDF values means that term is more discriminative in a particular document.

Latent semantic analysis (LSA) can be defined as "a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text."[LFL98] It is an automatic approach, that unlike others, takes raw data as input. It applies Singular Value Decomposition (SVD) to a previously created matrix in which cells contains the *weight* of a word in a piece of text inserted in a column. Turney [Tur01] proposed an algorithm called PMI-IR that was able to achieve better scores than LSA, where it assigns a score to a "choice" and then picks the "choice" that is able to maximize the score. It makes use of Pointwise Mutual Information (PMI).

Co-occurrence analysis is the analysis between multiple occurrences within a certain part of a text [Won09]. Its aim is to look within the text for certain terms that usually appear together with the goal of finding similar terms and relations [WLB12]. A good example on methods used for co-occurrence analysis are similarity measures [EW09] such as the cosine similarity which equals the ratio between the times two items appear together and the mean regarding the time each item is shown. The Jaccard Index unlike the previous method instead of the mean, uses the number of times at least one of the items is seen. Another measure is rank correlations such as Pearson and Spearman coefficient [Sho10]. The former equals the ratio of the covariance between two objects by the product of the their standard deviations. The later is a "rank-based version" of the first which means that is based on the ranked values of each object instead of raw data.

Clustering was briefly explained before, during the definition of Concept Learning. The process of forming and finding concepts can be regarded as a clustering problem [LP+04] since we predict that words in the same group provide similar context. Tan et al. [TSK05] provided an overview on some clustering techniques such as K-means. In this technique, the user defines the value of K which represents the number of clusters. These clusters will then be associated with a centroid that represents the center point of the cluster and are initially chosen at random. After assigning the initial centroid, the other points will be associated with the one closer to them, followed by the recalculation of the centroid in each cluster. This process is repeated until there are no changes to the clusters. Density-based spatial clustering of applications with noise (DBSCAN) [Mar] starts at a random point and gathers every point that is reachable from him, if no points are

reachable, the algorithm picks the next point as a starter. The algorithm may also split or merge different clusters based on their relation towards the input value ε.

3.2.2 Linguistic Techniques

Natural Language Processing is usually at the base [Won09] of linguistic techniques which can be used in most tasks of the ontology learning process.

Noisy text can be defined as any difference from the original text and consists of errors such as *lexical* (missing or additional characters) and *syntactic* (missing or additional words) [Sub10]. Different approaches regarding **Noisy text analytics** have since been proposed. Clark developed a tool that is able to process noisy text with a large quantity of synonyms [Arv02]. Another approach was proposed by Tsao et al. [TW09] and can be divided in two components: one is the use of hybrid n-grams that work as the standard when looking for errors in a learner production and the second is the use of the knowledge that came from the previous step by an error detection and correction algorithm.

Part-of-Speech tagging/Sentence analysis techniques consist of splitting a text into tokens to remove punctuation and then apply a marker based on its context [Spe13], this will then be used for further analysis [Won09]. Examples are *TreeTagger* which makes use of binary decision trees to calculate the transition probability [Sch94] and *Natural Language Toolkit (NLTK)* which is used by universities as a learning tool for NLP [BKLB08].

Seed Words/Terms are usually used as a base for some information extraction systems, such as ExDisco which takes as an input a set of seed words provided by the user, that will later be expanded by the system [YGTH00]. Hwang [Hwa99] built an approach for the creation of dynamic ontologies where the input are also seed words inserted by the user which will allow the extraction of relevant concepts and before each iteration may also generate new seed words.

Subcategorization frame can be defined as "a statement of what types of syntactic arguments a verb (or adjective) takes, such as objects, infinitives" [Man93]. Faure and Nédellec [FN98] developed a system for ontology learning which uses subcategorization frames as a structure for semantic knowledge which allows parsing of text by mapping these to grammatically parsed clauses.

Semantic Lexicon consists of a dictionary of words and the relations between each of them. Semantic lexicon offers similarities to lexical ontologies [Bie05] since both associate terms to concepts and keep similarities between terms. They are very popular to Ontology Learning [WLB12] as they allow access to compilations of concepts which are organized with respect to their similarities, which will then allow the discovery of terms that can then be used to create concepts. They can also be of two types — general such as HaGenLex [HHH+17] or WordNet [Mil95] and domain oriented such as the Specialist Lexicon [Pro99].

Syntactic Structure analysis / Dependency analysis is an approach that uses the syntactic structure for term extraction. This technique [LHC11] considers compound terms, which are used to represent concepts and are usually more specific and multi-worded, to be hyponyms of single worded terms. The head-modifier principle [HCA05] makes use of an element addressed as *head* that describes a variety of terms that are hyponyms. This is very helpful since it is not only domain

independent but also works "universally" allowing it to be use in a variety of languages as seen in the work of Hippisley et al. [HCA05] where the head-modifier principle was applied to the Chinese language.

3.2.3 Logic Techniques

As seen in Section 3.2, Logic techniques are usually adopted in the later stages of the ontology learning process regarding relations and rules. They consist of *logical inference* and *inductive logic programming* [WLB12]. Since they are out of the scope of this work, they will not be analyzed.

3.3 Existing Learning Systems

In recent years different tools have been developed. Most utilize hybrid approaches based on the different techniques seen in Section 3.2, with the aim of creating an ontology. In this section we will provide an overview on the techniques used by the tools that are related to keyword extraction and the formation of concepts.

3.3.1 ASIUM

Faure et al. [FN99] developed ASIUM, a system that learns ontologies from technical texts. ASIUM utilizes a syntactic parser named SYLEX which provides the semantic knowledge of texts. The authors have also developed a tool, that once paired with SYLEX extracts the "instantiated syntactic frames" from every sentence. An example for the sentence "Bart travels by boat" is provided by the authors [FNR] and the resulting instantiated syntactic frame is **<to travel><subject><Bart> <by><boat>** Since semantic acquisition is not influenced by ambiguities, when the last occurs, all syntactic frames are kept. The approach relies on the assumption that words share the same concept if they appear after the same preposition with the same verbs, which works well for technical domains where the vocabulary is more restricted. Clusters are used as support for the ontology and are created from instance frames that were previously extracted. These clusters are a list of the head words found using the assumptions described previously and are usually linked to their frequency. During the creation of concepts, similar clusters are aggregated using clustering methods; at the same time syntactic terms are being learned so when a concept is created it updates the syntactic frame. When the frames are reviewing text, if a sentence does not match, it must be either reformulated or a new concept may be added;the frame must then be improved to include this concept.

3.3.2 OntoLearn

Velardi et al. [VNCN05] proposed OntoLearn, an Ontology Learning system that makes use of both statistical and linguistic techniques. An overview of the steps performed by the system can be seen in Figure 3.3.

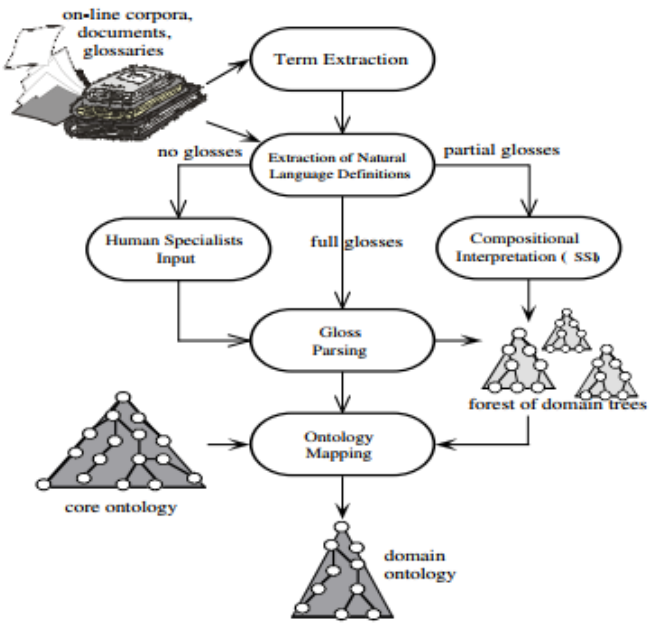


Figure 3.3: An overview of the different tasks in OntoLearn. Extracted from the work of Velardi et al. [VNCN05]

Statistical techniques are used for term extraction by selecting terms that are regularly found in documents related to that specific domain. Based on the extraction of terms or online glossaries it searches for natural language definitions and then, depending on if definitions have been found, it applies filtering algorithms to remove the ones that do not fit the interest of the domain. A Part-Of-Speech tagger such as *TreeTagger* is then applied to identify the different tokens inside a sentence, which will then be used to identify the main noun with the use of regular expressions.

If no definition was previously found for certain terms, namely, there is not a definition for a multi-worded term, but there are definitions for each term then it extracts those definitions from WordNet, and uses a *a word sense disambiguation algorithm* to select the appropriate concepts, followed by a machine-learning algorithm to identify the correct relations between this concepts. Following this step, a grammar will be created which will then be used to generate the natural language definitions. If the previous steps can not be done a domain expert should provide a definition for the terms.

3.3.3 OntoLearn Reload

OntoLearn Reload [VFN13] is the evolution of OntoLearn. It kept the first step regarding the extraction of terms, but now it does not rely on WordNet, which means that it does not depend on the English language. It also removed the need to use a word sense disambiguation algorithm to structure the taxonomy. Instead, it generates a hypernym graph based on the textual definitions extracted from the corpus and the Web.

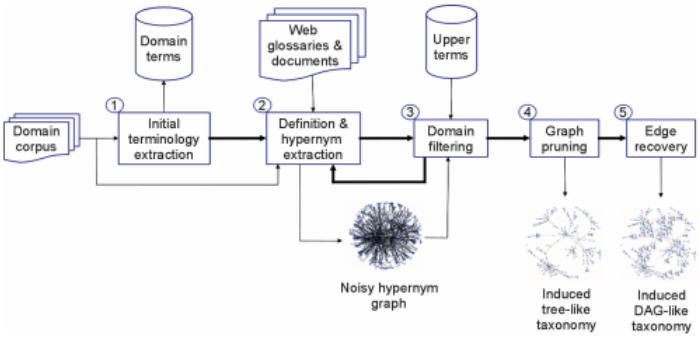


Figure 3.4: An overview of the different tasks in OntoLearn Reloaded [VFN13]

For the extraction of terms, it uses a tool that was also developed by the creators named termextractor[1] which produces a domain that may contain single and multi-worded terms. A classifier is then applied to each term to select proper definitions from the available sentences to obtain the hypernyms of that term. It is similar to OntoLearn as it applies a method to remove the definitions that are not in the interest of the domain. With the definitions left, it populates the graph with the hypernymy relations calculated before. Since the graph may contain certain cycles, it utilizes a weighting strategy to generate an optimum graph.

3.3.4 OntoGen

Fortuna et al. [FMG06] created OntoGen, a semi-automatic ontology learning system that helps the user by proposing terms based on keyword extraction from different domain documents.

OntoGen utilizes two different methods for discovering topics within texts. First is *Latent Semantic Indexing* (LSI) which is used in order to extract the background knowledge from documents. It is able to detect similar meaning words with the use of Singular Value Decomposition (SVD) and bag-of-words. The second is k-means, which was already explained in Section 3.2.1. In order to help the user interpret the different topics, the tool also applies two methods for keyword extraction [FGM06]: extraction using centroid vectors and Support Vector Machine (SVM). In the first, the important keywords are the words that have the biggest weight in the vector. As SVM works by using feature extraction to train a classifier and then it classifies the document by multiplying the previously computed bag-of-words by the SVM vector [Forb]. If the result is above a certain threshold, the document is considered positive, which means that, it is relevant to the category and the words with higher values are considered the most important. The difference between the two is that SVM contemplates context [FMG06], for example, if a category is named "computer" and we are interested in deciding what the keywords for a subcategory are, the first method would only take into account the keywords relevant to this subcategory which means that "computer" would be considered important despite we already knowing that it is the main category and our interest here is to find what makes each category differ from each. The second method takes this into account and uses every category to learn the most important words.

[1] http://lcl.uniroma1.it/termextractor

During the creation of the ontology, the previous methods will give suggestions in the various tasks, but ultimately it is the decision of the user to use them or not.

3.3.5 Syndikate

Syndikate was proposed by Hahn and Romacker [HR01] and stands for *Synthesis of Distributed Knowledge Acquired from Texts*. It was applied to German technical documents regarding the areas of information technology and a medical sub-domain. It the only tool that only applies linguistic techniques for every task.

Syntactical structure analysis is executed using a dependency grammar referred as *Lexicon*. The aim is to capture binary valency constraints between an element such as a noun, and possible modifiers. In order to do this, certain restrictions have to be satisfied: compatibility of morpho syntactic features, word order and semantic criteria. When handling pronouns anaphora resolution is used.

Using semantic templates that were previously defined, every term in the dependency graph is matched with a concept from the domain and are also used to express the text knowledge base [Won09]. This base is simply a representation of the texts with an explanation.

3.3.6 CRCTOL

CRCTOL was developed by Jiang and Tan [JT05] and stands for *Concept Relation Concept Tuple based Ontology Learning*. The three main components of CRCTOL can be see in Figure 3.5 and consists of *Natural Language Processing*, which includes tools such as POS tagger and a syntactic parser; *Algorithm Library* that includes the algorithms responsible for extracting key concepts within a collection, a rule based algorithm for relation extraction from key concepts and a rule mining algorithm to build the ontology; *Domain Lexicon* which is created by the user and includes terms from the domain that will be used by the Natural Language Processing component to analyze documents.

In the concept extraction phase, the authors applied a method which reduces the chance of loosing essential concepts. After extracting multi-worded terms, it selects "candidates" followed by the calculation of the linear combination of each. The one with the highest value will then be used to form a list of concepts. If during the calculation of the term frequency of each single-worded term that is on the list as a *head*, there is a term that has a value above a certain threshold, it is added to the list previously created.

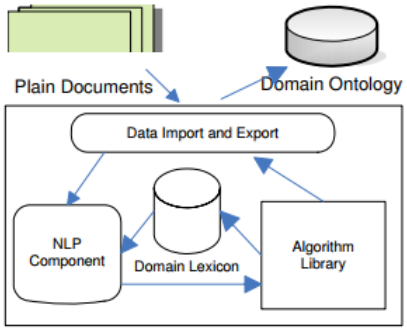


Figure 3.5: An overview of the core components of CRCTOL [JT05]

3.3.7 OntoGain

OntoGain [DZP10] is another ontology learning tool and depends on multi-worded extraction from unstructured text. The use of multi-worded terms allows the system to be more efficient since the representation is tighter and also allows better coverage of the domain. During the preprocessing phase, OpenNLP, Part-Of-Speech tagging and tokenisation were used while WordNet Java Library(JWNL) was used for lemma information. During concept extraction C/NC-value is used for the extraction of multi-worded terms. C-value consists of the relation between the frequency of occurrence of a word sequence and the frequency that the same sequence might be part of a bigger term inside a text. NC-value improves C-Value by applying weights to the previous terms.

Then, from each term, a clustering method is applied that considers each term to be a cluster, based on similarities. These clusters will then be merged until there is only one.

3.3.8 TERMINAE

TERMINAE [BSC99] was created with certain requirements in mind: the use of *Linguistic-based methods*; *typology of concepts* to allow proper maintenance of the ontology by allowing the differentiation between the different modeling choices; inconsistency and incoherence prevention by using *formalities* and ontology documentation to facilitate the verification of the conceptualization.

TERMINAE starts by applying a term extractor called LEXTER which offers the user a list of possible term. This user will then need to select the relevant terms and define the possible meanings of each one. A natural language must then be provided and translated to a formalism. If valid, the new concept may then be inserted.

3.3.9 Text-to-Onto

Maedche and Volz [MMV01] created Text-to-Onto, a tool which uses both data mining and natural language processing in order to help the user during the process of creating an ontology. It allows as an input either semi-structured (HTML) or unstructured data (PDF) and applies pre-processing techniques as a way to remove noise from texts before applying natural language techniques. The authors affirm that this process greatly improved the process.

Following the previous step, a text processor for the German language is applied named Saarbruecken Message Extraction System (SMES) which is responsible for producing syntactic structures and dependencies, followed by an analysis on these structures to identify terms. For the creation of concepts [WLB12] it may either use a domain-oriented lexicon as opposed to other ontology learning tools or co-occurrence analysis.

3.3.10 Text2Onto

Text2Onto [CV05] is the successor of Text-to-Onto and was created by Cimiano and Volker, the first being the creator of the Ontology Learning Layer Cake.

The model was developed with a few issues in mind — the need to restart the ontology creation from the start if the corpus has been changed, the limited interaction with the users when they should be having a central role in the creation, and the fact that most tools are restricted to a certain format by using specific ontology models. The tools solve this issues by adding a Probabilistic Ontology Model (POM) and Data-driven Change discovery.

A POM works as a container for objects that were learned and each one has a *calculated probability* which will allow the user to conclude if the object should be added to the ontology. Each object also includes a pointer to the documents from where it came. Data-driven Change discover is the process that allows the ontology to be updated instead of compiled from the start.

During the preprocessing task, Text2Onto uses the Gate framework for sentence detection, tokenization, POS-tagging and JAPE pattern rules, these are language specific rules since Text2Onto does not have support for every language. The algorithms for the creation of concepts are Relative Term Frequency, TF-IDF, Entropy, C/NC-value.

3.4 Summary

In this chapter we introduced the ontology learning area which is where our main goal is inserted. Despite the improvements that have appeared in areas such as information retrieval and machine learning, this is still a semi-automatic process. We then described the sequence of tasks during a creation of an ontology and the different approaches and systems that are currently used in the process. We will later analyze them so that we can design our own system to be used during the creation of an ontology for Dendro.

Chapter 4

Comparison of keyword extraction approaches

4.1 Introduction

In the previous Chapter we provided an overview of the ontology learning area, including techniques and tools typically used during the semi-automatic creation of an ontology. Here we will evaluate some state of the art Automatic Keyword Extraction (AKE) methods, namely TF-IDF and RAKE, a promising new AKE method named Yake! and C-value which is a state of the art method in Automatic Term Recognition (ATR). The ones that offer the best results will later be offered as an option during the term extraction phase of the module to be developed.

We were unable to test the ontology learning systems presented in Section 3.3, due to their unavailability.

4.2 Methods

The methods were chosen based on multiple factors. TF-IDF and C-value because they are not only considered state of the art but are also implemented in existing state of the art ontology learning systems, such as, Text2Onto [CV05] and OntoGain [DZP10]. Yake! [CMP+18a] because it is a keyword extraction method that is being developed within the same institution as this project and has also shown promising results and RAKE [RECC10] because it is one of the most well-known and used unsupervised keyword extraction methods and will allow a fair evaluation against Yake!. Another factor that made us consider these methods is that they are either already available as packages for the programming languages we are using to develop our tool or will be easy to port.

4.2.1 TF-IDF

TF-IDF is an weighting extraction method currently used by many ranking methods, that although simple, has proved to be strong and hard to beat by more complex methods [Rob04].

TF stands for term frequency and consists of the numbers of times a term occurs within a document. Term frequency is given by the following formula [G+16]:

$$tf(t,d) = \frac{f_{(t)}}{n} \tag{4.1}$$

- tf(t,d) is the term frequency value, with t representing the term and d the document
- f(t) is the term frequency
- n is the number of the terms in the document

Although the term frequency of a term is a simple and efficient method of representing its importance in a given document, common words that usually appear multiple times but are less relevant, end up with a higher score. Certain methods like C-Value [FAM00] make use of a stopword list during preprocessing in order to remove these kind of terms. To solve this problem in the this context, inverse document frequency (IDF) was proposed. A word with low IDF means that it occurs throughout multiple documents, meaning it is not relevant for the topic [LLL08]. The formula for IDF is:

$$idf(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{4.2}$$

- idf(t) is the inverse document frequency of term t
- d is the number of documents that contain the term t
- D is the number of documents in the corpus

And the complete TF-IDF formula is:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \tag{4.3}$$

TF-IDF is a method that favors terms that appear multiple times but in a short number of documents throughout the corpus. By having a high TF-IDF score, a term can be considered relevant and used to summarize the document in which it is present [G+16].

4.2.2 C-Value

In Section 3.2 we provided an overview of both statistical and linguistic techniques used in ontology learning, and while the previous method (TF-IDF) is considered a statistical approach, C-Value has both statistical and linguistic aspects, making it an hybrid approach to term extraction.

The linguistic aspect of the method consists of three steps: POS-tagging, application of a linguistic filter to the corpus in order to remove unwanted terms and the use of a stoplist, which can be altered to fit the needs of the user. It is used in order to further exclude terms that passed the previous filter but contain either common or irrelevant words [FAM00]. The authors [FAM00] of the method compared three filters during their work: (Noun)+Noun, (Adj | Noun)+Noun and

((Adj|Noun)+|(Adj|Noun)*(NounPrep)?)(Adj|Noun)*)Noun. The use of these filters vary according to the needs of the user. A closed filter such as the first will most likely present a better precision score, while a more open filter, such as the second will provide better recall since it recognizes more terms. The authors [FAM00] also compared these three filters, in which the second filter provided the best scores, followed by the first. Following their results, we decided to also do our own evaluation using these two filters.

Following these three steps, the statistical component are applied. These steps are mainly dependent on four characteristics of the terms [FAM00], namely, the total frequency of occurrence of the candidate in the corpus, the frequency of the candidate as part of longer candidates (nested frequency), the number of these longer candidates and the length of the candidate (in number of words). An example of these characteristics are the terms “aerodynamic drag coefficient" and “aerodynamic drag". The term “aerodynamic drag coefficient" has a frequency of 5, has never appeared inside a longer term and has a length in number of words of 3. The term “aerodynamic drag" appears only inside the term “aerodynamic drag coefficient" and has a frequency of 8. This means that “aerodynamic drag" has a frequency of 8, 2 as the length of the term, a nested frequency of 5 and 1 as the number of terms that contain it.

The C-Value formula based on these characteristics is [G+16]:

$$C-value(a) = \begin{cases} \log_2 |a| \cdot f(a), & \text{if a is not contained in a longer term} \\ \log_2 |a|(f(a) - \frac{1}{P(T_a)} \sum_{b \in T_a} f(b)), & \text{otherwise} \end{cases} \tag{4.4}$$

- a is the candidate term
- f(a) is the frequency of occurrence in the corpus
- T_a is the set of terms that contain a
- f(b) is the total frequency of the term that contains a
- $P(T_a)$ is the number of terms

The combination of both of linguistic and statistical information is said to improve the precision of the extraction. Frantzi, et al. [FAM00] have also proposed an augmented version of C-Value, named NC-Value which adds context to the terms extracted previously in order to improve their distribution in the final list, with the relevant terms moving closer to the top. Unlike C-Value, we could not find a usable implementation of NC-Value in order to run these tests, although depending on the evaluation of C-Value, it may latter be implemented in the tool.

4.2.3 RAKE

Rapid Automatic Keyword Extraction (RAKE) is an unsupervised and both domain and language independent keyword extraction method for single documents [RECC10]. As previously stated, it is one of the most well known keyword extraction methods.

RAKE takes as input a stopword list, a set of phrase delimiters and a set of word delimiters [RECC10]. Based on this input, RAKE begins the extraction of keywords by transforming the text into a set of candidate keywords.

After identifying every candidate, a keyword is calculated based on several metrics, word frequency (frequency(w)), word degree (degree(w)), which is, the sum of the length of all the phrases where the word occurs and ratio between degree and frequency (degree(w)/frequency(w)). Words that tend to appear within other longer candidates are favored by degree(w) while words that just occur frequently, regardless of where, are favored by frequency(w). After every candidate has a score associated, the top scoring are then extracted.

The fact that RAKE is both efficient and simple makes it suitable for large collections of documents [RECC10].

4.2.4 Yake!

Yet Another Keyword Extractor (Yake!) is a keyword extraction method for single documents, and like RAKE is unsupervised, meaning it does not rely on dictionaries or is trained beforehand [CMP+18b].

This method has four main components: text preprocessing, feature extraction, individual term weighting and candidate keywords generation [CMP+18a]. During preprocessing the text is tokenized into terms. Then, a set of five features are considered for the terms: Casing; Word Position; Word Frequency; Word Relatedness to Context; and Word DifSentence. Casing is related to either words that start with capital letters or acronyms, since there is the assumption that these words are usually more relevant. Word position is considered since there is the belief that relevant words usually concentrate towards the beginning of a document [CMP+18b]. Word frequency is related to the number of times a word occurs in the text. Word Relatedness to Context consists of the number of terms that occur to either side of the candidate term, with the more terms co-occurring with the candidate, the less relevant this last one is. The last feature, Word DifSentence, is related to how often a word appears inside different sentences. In the next step, each feature is combined into a single formula where the lower the score, the more important the word is. The formula is:

$$S(w) = \frac{W_{Rel} * W_{Position}}{W_{case} + \frac{W_{Freq}}{W_{Rel}} + \frac{W_{DifSentence}}{W_{Rel}}} \tag{4.5}$$

Following the previous step and considering the fact that keywords may be comprised of multiple words, sequences between 1 and 3-grams will be generated where each candidate will have a final S(kw) value assigned. In the end, the system removes similar candidates and returns a list consisting of the most relevant keywords [CMP+18b]. The formula for the last step can be seen below:

$$S(kw) = \frac{\prod_{w \in kw} S(w)}{TF(kw) * (1 + \sum_{w \in kw} S(w))} \tag{4.6}$$

4.3 Datasets

In order to evaluate the previous methods, four different datasets used in keyword extraction tasks were selected. These datasets were chosen because they are made up entirely of scientific documents written in English, which is also the kind of input our tool will take. The datasets are Fao30 [Med09, MW08] which consists of 30 agricultural documents obtained from the United Nations Food and Agriculture Organization (FAO)[1] indexed by professional indexers with terms from Agrovoc[2]; the dataset that was used during SemEval2010[3] which contains 284 scientific articles with keyphrases that were chosen by both readers and authors [KMKB10]; Nguyen2007 [NK07] which contains over 200 documents where each document has a length between 4-12 pages and each keyphrase from the documents were assigned by student volunteers; and finally NLM500 which contains 500 PubMed[4] documents with MeSH[5] terms.

4.3.1 Datasets preparation

Apart from Nguyen2007, all datasets come in a single folder with two files for each document, one containing the scientific paper and the other containing a list of manually assigned keywords. This made it easy to develop a small tool in order to compare the list of keywords with the ones extracted from each method. Nguyen2007 contained a folder for each document with another folder inside containing the assigned keywords. This forced us to copy the files from each of these folders to a single folder in order to match the other datasets being used.

Also, the implementation of TF-IDF that we used demanded the input files to have POS-tags associated with each word in order to allow the extraction of keywords. To do this, we applied a POS-tagger from a natural language processing software named Stanford CoreNLP[6]. Since this software will also be used in the preprocessing phase of our tool, we will only go into detail about how it works in the next Chapter. An example of the addition of POS-tags the datasets can be seen in Fig 4.1.

[1] http://www.fao.org/home/en/
[2] http://aims.fao.org/vest-registry/vocabularies/agrovoc-multilingual-agricultural-thesaurus
[3] http://semeval2.fbk.eu/
[4] https://www.ncbi.nlm.nih.gov/pubmed/
[5] https://meshb.nlm.nih.gov/
[6] https://stanfordnlp.github.io/CoreNLP/

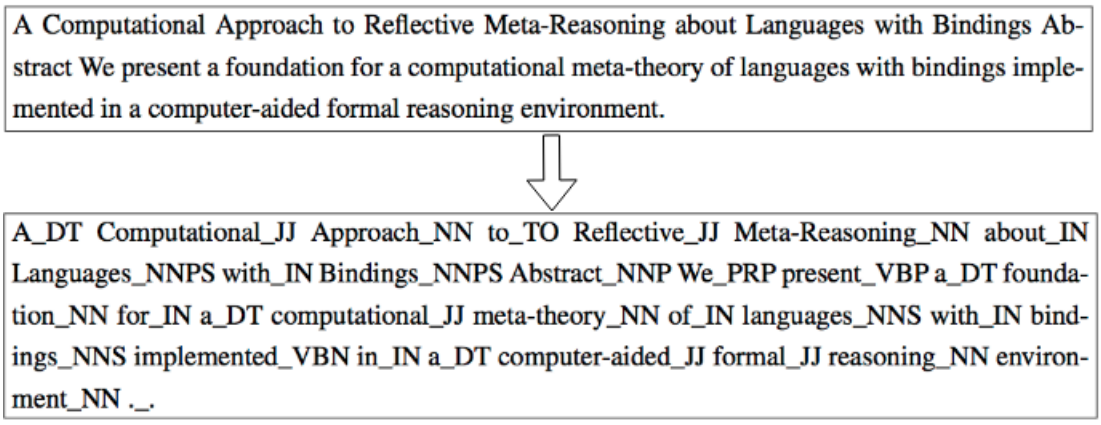


Figure 4.1: Example of the addition of POS-tags to text

4.4 Evaluation

In order to evaluate the methods previously described with these datasets we have decided to use the same parameters as the Sem Eval 2010 task 5 competition [KMKB10].

4.4.1 Evaluation Method

The precision of each method was calculated for the top 5, 10 and 15 extracted keywords. We also only consider a match between an extracted keyword and a manually assigned one when there is an exact match.

The ranking was based on three metrics [Pow11]: Precision, which is defined as the number of true positives (Tp) over the number of true positives (Tp) plus the number of false positives (Fp). Recall, which consists of the number of true positions(Tp) over the number of true positives (Tp) plus the number of false negatives (Fn) and F-measure which combines both precision and recall and is the harmonic mean between these. To simplify, precision is the ratio between the correctly extracted terms and the number of extracted terms, while recall is the ratio between the correctly extracted terms and the number of number of correct terms in the entire collection. The formulas for all three can be seen below:

$$P = \frac{T_p}{T_p + F_p}; R = \frac{T_p}{T_p + F_n}; F = 2 \cdot \frac{P \cdot R}{P + R} \tag{4.7}$$

4.4.2 Tools used

For Yake!, we have used the implementation[7] made available by the authors, which takes as an input a .txt file and returns the top 20 keywords of that file [CMP+18a, CMP+18b]. For RAKE we have decided to use a Python-based implementation[8] that was also used by the Yake! authors

[7] https://pypi.org/project/yake/
[8] https://github.com/zelandiya/RAKE-tutorial

during their own evaluation. As for TF-IDF we have used a package[9] that takes as an input a .txt file annotated with POS tags. Both the Rake and the TF-IDF implementation allowed the option to include the keyword file for each document in order to automatically calculate the average precision, recall and f-score for the dataset being used. However, these calculation required some modifications in order to fit the evaluation requirements. At last, for C-Value we used a Java based implementation[10] that makes use of the Illinois POS Tagger[11] for the linguistic part and has been previously tested on thousands of files by the creator.

4.4.3 Results

The results for the evaluation of the methods can be seen in the Tables 4.1, 4.2, 4.3 and 4.4.

The results for each dataset follow a similar pattern, in which, Yake! always has the best score followed by each of C-Value filters, namely, (Adj|Noun)+Noun and Noun+Noun. The difference in values between the top scoring C-Value filter and Yake! was between 1 and 3% depending on the dataset being used.

The implementations of both Rake and TF-IDF had much lower results. Apart from SemEval, where TF-IDF reached an average precision and recall of around 3.6%, both of them remained under 1%.

	Top 5			Top 10			Top 15		
	P	R	F-m	P	R	F-m	P	R	F-m
Rake	0.1	0.1	0.1	0.7	0.4	0.51	0.55	0.47	0.51
Yake!	16.8	5.57	8.37	14.55	9.72	11.65	12.62	12.6	12.61
TF-IDF	2.13	0.69	1.04	2.86	1.86	2.26	3.66	3.56	3.61
C-Value Noun+Noun	11.39	3.86	5.77	9.14	6.09	7.31	7.76	7.69	7.72
C-Value (Adj|Noun)+Noun	15.98	5.27	7.93	12.46	8.32	9.98	10.49	10.44	10.46

Table 4.1: Results for the SemEval 2010 Task 5

	Top 5			Top 10			Top 15		
	P	R	F-m	P	R	F-m	P	R	F-m
Rake	0.1	0.1	0.1	0.19	0.26	0.22	0.19	0.41	0.26
Yake!	16.46	12.97	14.51	12.49	19.14	15.12	10.56	24.05	14.68
TF-IDF	0	0	0	0.24	0.31	0.27	0.41	0.82	0.55
C-Value Noun+Noun	10.81	8.31	9.4	7.75	11.69	7.17	6.41	14.01	8.8
C-Value (Adj|Noun)+Noun	14.45	11.55	12.84	11.2	17.26	13.58	9.03	20.26	12.49

Table 4.2: Results for Nguyen 2007

[9] http://www.hlt.utdallas.edu/ saidul/code.html
[10] https://github.com/Neuw84/CValue-TermExtraction
[11] http://cogcomp.cs.illinois.edu/page/software_view/POS

	Top 5			Top 10			Top 15		
	P	R	F-m	P	R	F-m	P	R	F-m
Rake	0.08	0.03	0.04	0.1	0.08	0.09	0	0	0
Yake!	7.48	3.06	4.34	6.12	4.85	5.41	5.32	6.3	5.77
TF-IDF	0.52	0.18	0.27	0.74	0.51	0.61	0.8	0.84	0.82
C-Value Noun+Noun	3.08	1.12	1.64	2.36	1.73	2.0	1.97	2.14	2.05
C-Value (Adj|Noun)+Noun	3.2	1.19	1.73	2.48	1.87	2.13	2.11	2.39	2.24

Table 4.3: Results for NLM 500

	Top 5			Top 10			Top 15		
	P	R	F-m	P	R	F-m	P	R	F-m
Rake	0	0	0	0	0	0	0	0	0
Yake!	11.33	4.43	6.37	10.33	7.76	8.86	9.11	10.53	9.77
TF-IDF	0	0	0	0	0	0	0.24	0.26	0.25
C-Value Noun+Noun	10.0	3.67	5.37	8.0	5.76	6.7	6.89	7.94	7.38
C-Value (Adj|Noun)+Noun	10.67	4.0	5.82	8.33	6.17	7.09	7.11	8.21	7.62

Table 4.4: Results for Fao 30

4.5 Summary

In this chapter we provided a more detailed explanation of some well-know keyword extraction methods and some datasets in which those can be evaluated on. Yake! and C-Value were clearly the best methods evaluated and due to their proximity in both precision and recall values, we have decided to implement both into our tool. They will later be analyzed in the context of ontology learning.

Chapter 5

Dendro Keywords

5.1 Introduction

In the previous chapter we provided a comparison between some state of the art keyword and term extraction methods. Based on those results and pairing them with certain techniques described in Chapter 3 we have developed a module for Dendro in which a project administrator, either a curator or researcher, can extract concepts from existing files within the project. This is done in order to support the curator during the creation of an ontology.

In this chapter, we will describe the different phases of the tool development while also showing examples of its functionality. The tool was built based on the first steps that usually comprise an ontology learning tool, namely, preprocessing, term and concept extraction.

5.2 Proposed solution

The current process of creating an ontology for describing datasets in Dendro is a manual process which requires an interview between a curator and a research, after which, the curator will analyze the research data and propose an ontology which will ultimately be validated by the researcher.

In Fig 5.1 we can see a visual representation of this process, with the addition of our proposed module. The purpose of the tool is not to be used as a replacement for the process, but to actually be used as an auxiliary tool through the duration of it. The tool will allow curators to not only save time, but also offer concepts that he might have overlooked or found uninteresting at first.

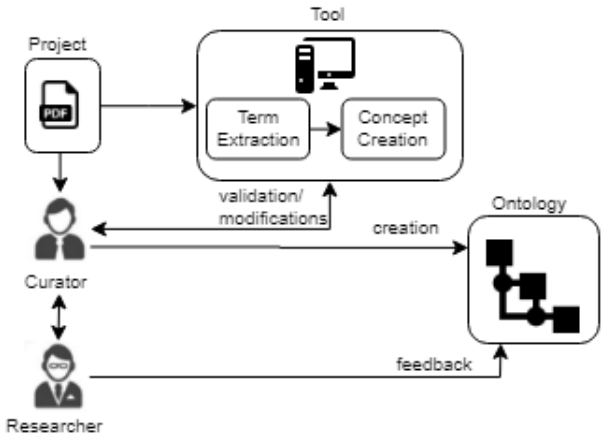


Figure 5.1: Addition of the module to the ontology creation process

A representation of the interaction between the different steps that will be explained in this chapter can be seen in Fig 5.2.

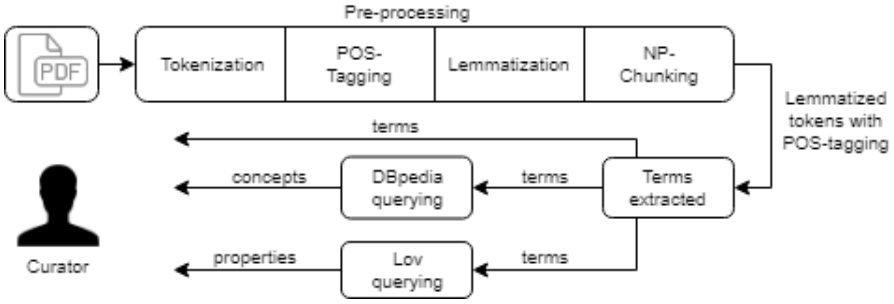


Figure 5.2: Workflow diagram [MLS18]

5.2.1 Preprocessing

Preprocessing is not only the first task, but it is also one of the most important in ontology learning systems. It is also very similar throughout the different tools that are currently available as seen in Chapter 3.3.

For every task in preprocessing we use Stanford CoreNLP. The choice was based in two different factors: existing comparisons between different NLP toolkits, such as the the one made by Ievgen Karlin [Kar12] in which CoreNLP was considered one of the most suitable NLP tools, and the availability of the tool for the technologies used in the implementation.

In our work, preprocessing can be split into five different steps: tokenization, sentence parsing, POS-tagging, lemmatization, and noun phrase extraction. We will use the sentence *"The transportation domain has gained significance in the context of climate change and energy savings."* [PMR+14] as an example in order to show the output from the different steps. An overview of the example starting with the sentence already split into tokens can be seen in Fig. 5.3.

Tokenization is the process of splitting text into words. Tokenization is always required before any type of preprocessing task [Tri13].

POS-tagging was briefly explained in Section 3.2 and consists of applying a tag to a token based on factors, such as its relation with adjacent words. The English tagger available in CoreNLP makes use of the POS-tags from Penn Treebank[1]. Penn Treebank contains a total of 36 tags depending on how the word shows up in the text, such as "NN" for singular nouns or "NNS" for plural nouns. In our case, we do not make a distinction between different type of nouns or adjectives. These tags will then define the output of the next steps.

The next step consists of reducing words to a common base form, such as, converting plural words to singular or different verb tenses to the verb base form. In order to do this, either a stemmer or lemmatizer may be applied. A stemmer is considered to be a more aggressive approach to the task, since it cuts the end of words based on different rules while the lemmatizer takes into account the morphological analysis of the word [Lar10]. In our approach we have decided to use a lemmatizer due to the existence of one in CoreNLP.

A noun phrase is a phrase that has a noun as its head. Noun phrase extraction consists of extracting noun phrases from each sentence. This is essential since we will use C-value which computes a score for noun phrases based on a number of characteristics. The noun phrases we extract have two requirements: need to contain at least one word and fit a linguistic filter. These filters are the ones used to test C-Value and consist of Noun+Noun or (Adj|Noun)+Noun, meaning that our terms consist of at least two words containing either all nouns or a mixture of adjectives and nouns [FAM00]. Also, there are no limits to size of the noun phrases, meaning they can go from a minimum of 2 to any number of words. As we can see in Fig. 5.3 there are three noun phrases, each containing a sequence of nouns, meaning they would fit either the first or the second filter.

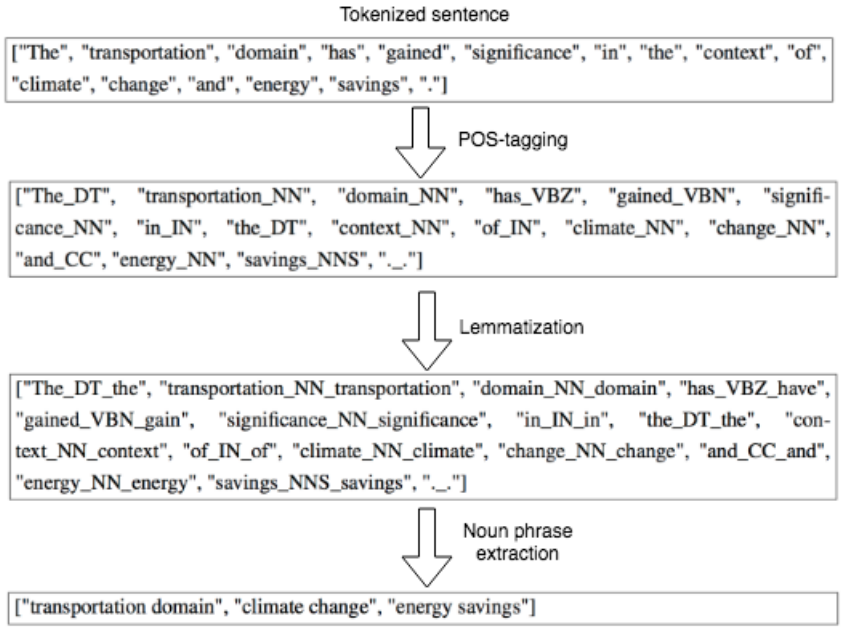


Figure 5.3: Example of the preprocessing steps

[1]https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

To implement CoreNLP in our work we have used a library[2], which is currently under development, and facilitates the use of CoreNLP in NodeJS[3]. This library provides access to the sentence splitting, tokenizer, POS-tagger and lemmatizer capabilities of CoreNLP. Based on the output of these tasks we are able to then extract the noun phrases in each sentence.

5.2.2 Term extraction

By analyzing the results obtained in Chapter 4, we have decided to implement both Yake! and C-Value. C-Value will be able to make use of the noun phrases extracted in the last step of the preprocessing phase, while Yake! due to not relying on any linguistic aspect of texts, apart from a stopword list, will only use the processed text. These term extraction methods will be further analyzed in the context of ontology learning in Dendro on the next chapter.

Despite using an existing implementation of C-Value during the keyword extraction evaluation, for our module we had to develop our own version from scratch due to the use of different technologies. As for Yake! we were able to use the implementation[4] made available by the authors. An example of the output of this implementation can be seen in Fig 5.4.


```
{
  "keywords": [
    {
      "ngram": "highly radioactive water",
      "score": 0.012872678741596309
    },
    {
      "ngram": "crippled nuclear plant",
      "score": 0.012872678741596309
    },
    {
      "ngram": "japan official",
      "score": 0.022599163614081593
    }
  ],
  "language": "english"
}
```


Figure 5.4: Output of the Yake! API for a text excerpt

5.2.3 Clustering

Earlier we explained that we would only focus on the term extraction and concept learning steps of ontology learning since the ontologies built for Dendro are lightweight and possess no relations between each of the descriptors. However, we have decided that although relations will not be included in the ontology, they may offer a better visual representation of the concepts being displayed by aggregating concepts that might possess similar meaning.

[2]https://github.com/gerardobort/node-corenlp
[3]https://nodejs.org/en/
[4]https://boiling-castle-88317.herokuapp.com/apidocs/#!/available_methods/post_yake_v2_extract_keywords

We have decided to implement a version of Agglomerative Hierarchical Clustering similar to the one existing in OntoGain [DZP10] and the one available in an ontology learning tool built for Swedish text [Bot15] since these have shown to provide better results than the methods used as comparision, such as, Formal Concept Analysis (FCA) [Dry09].

Agglomerative Hierarchical Clustering (HAC) is the name given to the bottom-up approach of hierarchical clustering [Lar10]. In a bottom-up clustering approach each item (in our case concept) starts as single clusters which are then merged based on similarity until only one remains. In order to evaluate the similarities between each pair of clusters we use the Group-average agglomerative clustering (GAAC), which calculates the average similarity between every concept within the clusters being merged [Lar10].

In order to calculate the similarity between two concepts we use the Lexical Similarity method, which is based on the following assumptions [NSA02]: terms sharing the same head are assumed to be (in)direct hypernyms of the same concept (e.g. hybrid vehicle and electric vehicle are both vehicles) and when we have a term that contains other, the first is considered a specialization (e.g. aerodynamic drag and aerodynamic drag coefficient). The formula for the Lexical Similarity is [Dry09]:

$$LS(t1,t2) = \frac{h_1 \cap h_2}{h_1 + h_2} + \frac{c_1 \cap c_2}{c_1 + c_2} \tag{5.1}$$

On the left side we have the number of shared heads over the number of total heads in the term. On the right, the number of matching combinations over the number of total combinations. Table. 5.1 shows an example on Lexical Similarity calculations.

	Term	Combinations
1	electric vehicle	electric, vehicle, electric vehicle
2	hybrid vehicle	hybrid, vehicle, hybrid vehicle
3	hybrid electric vehicle	hybrid, electric, vehicle, hybrid electric, electric vehicle, hybrid electric vehicle
LS(1,2) = 0.67; LS(1,3) = 0.83; LS(2,3) = 0.72		

Table 5.1: Lexical Similarity

For the clustering task we have decided to build our approach on top of an existing implementation[5]. We used the basic functionalities of the package while defining our own version of the distance calculation and also the addition of certain conditions.

5.2.4 DBpedia and LOV querying

Dendro Keywords uses DBpedia as a method to provide a meaning to a specific term by associating a term with an existing resource on DBpedia. We chose DBpedia because it provides the content of Wikipedia in a structured format, allowing more sophisticated queries [ABK+07].

[5] https://www.npmjs.com/package/hierarchical-clustering

We are currently querying DBpedia in order to associate a concept to each extracted term. But since these concepts are usually resources and since one of our main goals is to suggest properties as candidate descriptors, we have also decided to make queries to Linked Open Vocabularies (LOV). Started in 2011, the Linked Open Vocabularies initiative[6] is an “innovative observatory of the semantic vocabularies ecosystem" [VAPVV17]. The goal of LOV is to facilitate the reuse of properly documented vocabularies in the Linked Data ecosystem. LOV allows the search for vocabularies terms, let it be classes or properties based on a certain domain. We chose LOV based on this featured, allied with the always increasing number of vocabularies available [VAPVV17].

In order to match a term and a DBpedia resource we used a tool named DBpedia Lookup[7]. This tool allows us to make a REST query to DBPedia with a term in order to obtain a label, URI and description for a resource matching that term. An example of the output of the query for the term "machine learning" can be seen in Fig 5.5.


```
{
    "results": [
        {
            "uri": "http://dbpedia.org/resource/Machine_learning",
            "label": "Machine learning",
            "description": "Machine learning, a branch of artificial intelligence, is
a scientific discipline concerned with the design and development of algorithms that
allow computers to evolve behaviors based on empirical data, such as from sensor data
or databases. A learner can take advantage of examples (data) to capture
characteristics of interest of their unknown underlying probability distribution. Data
can be seen as examples that illustrate relations between observed variables."
        }
      ]
}
```


Figure 5.5: Output of a DBpedia Lookup query for the term machine learning

The only problem that we found when using the DBpedia Lookup is the absence of order in the results, which means that even if one of the results is an exact match against our search term it may appear on the bottom. In order to solve this, we have decided to compute the Dice coefficient between the search term and the label of each result in order to return the most similar. The Dice coefficient is a statistical method used to compare the similarity between two strings [DSM99]. This method has shown good results since the DBpedia label is often very similar to our search term.

In order to make our queries to LOV we have used their own API[8] which allows us to search for a term, while defining the type of output we want, in our case, property. An example of a query to LOV can be seen in Fig 5.6.

[6] http://lov.okfn.org/dataset/lov/
[7] https://github.com/dbpedia/lookup
[8] http://lov.okfn.org/dataset/lov/api

```
{
  "prefixedName": [
    "dbpedia-owl:vehicle"
  ],
  "metrics.reusedByDatasets": [↔],
  "vocabulary.prefix": [
    "dbpedia-owl"
  ],
  "metrics.occurrencesInDatasets": [↔],
  "uri": [
    "http://dbpedia.org/ontology/vehicle"
  ],
  "type": "property",
  "score": 2.5481486,
  "highlight": {↔}
},
```


Figure 5.6: Excerpt of the output from a LOV query for the term "vehicle"

5.3 System Architecture

Based on the described solution we have developed a module for Dendro. This module is both available as an application programming interface (API) and built in on Dendro within the project administration page.

5.3.1 API documentation

Despite our main goal always being to provide a user interface for Dendro Keywords, we have first developed it as an API, as in the future it can enable seamless interaction with external systems.

There are currently 6 methods available in the API which can be seen Table 5.2.

Method	Route	Definition
POST	/keywords/processextract	Preprocess documents and extract terms
POST	/keywords/preprocessing	Preprocess documents
POST	/keywords/termextraction	Extract terms from processed documents
POST	/keywords/clustering	Applies the agglomerative hierarchical clustering
POST	/keywords/dbpedialookup	Search for concepts in DBpedia based on terms
POST	/keywords/lovproperties	Search for properties in LOV based on terms

Table 5.2: Available API methods

Preprocessing method

Receives in the request body the text of the document it is supposed to process and returns the processed text, a list of all the noun phrases extracted and a list of every word, their lemma and POS-tag.

Request content:

```
{
  text: [
        {text: "Urban and suburban centers rely upon their
           transportation..."},
        {text: "This paper intends to analyze the performance of an
           electric bus..."}
  ]
}
```

Response content:

```
{
  text: "Urban and suburban center rely upon they transportation..."
     ,
  result:[
    {word: "Urban", pos: "NNP", lemma: "Urban"},
    {word2: "suburban":, pos: "JJ", lemma: "suburban"}
  ],
  nounphraselist: ["suburban center", "electric bus"]
}
```

Term extraction method

Depending on the chosen extracted method, it receives either the processed text (Yake!) or both the processed text and list of nounphrases (C-Value). This is due to Yake! not relying on any natural language processing techniques. It returns a list of every term ordered by method score.

Request content:

```
{
  method: "cvaluejj",
  text: "Urban and suburban center rely upon they transportation..."
     ,
  nounphraselist: ["suburban center", "electric bus"]
}
```

Response content:

```
{
  keywords: [
      {word: "suburban center", score: 21},
      {word: "electric bus", score: 14}
  ]
}
```

Clustering method

Receives a list of terms and returns these terms aggregated as clusters based on their lexical similarity.

Request content:

```
{
  keywords: [
      {word: "suburban center", score: 21},
      {word: "electric bus", score: 14}
      {word: "hybrid bus", score: 11}
  ]
}
```

Response content:

```
{
  clusters: [
      cluster: [
          "electric bus", "hybrid bus"
        ],
        cluster2: [
          "suburban center"
        ]
    ]
}
```

Preprocess and extract method

Combines both of the *Preprocessing* and *Term extraction* methods by taking as input the extraction method and the documents and returning a list of ranked terms.

Request content:

```
{
  method: "cvaluejj",
  text: [
        {text: "Urban and suburban centers rely upon their
           transportation..."},
        {text: "This paper intends to analyze the performance of an
           electric bus..."}
  ]
}
```

Response content:

```
{
  keywords: [
      {word: "hybrid vehicle", score: 54},
      {word: "aerodynamic drag", score: 36}
  ]
}
```

Dbpedia querying method

Receives a list of terms and returns the DBpedia resources related to those terms.

Request content:

```
{
  keywords: [
      {word: "hybrid vehicle", score: 54},
      {word: "aerodynamic drag", score: 36}
  ]
}
```

Response content:

```
{
  result: [
      {searchterm: "hybrid vehicle", dbpedialabel: "Hybrid vehicle",
          dbpediauri: "http://dbpedia.org/resource/Hybrid_vehicle",
         dbpediadescription: "A hybrid vehicle is a vehicle that
         uses two or more distinct power sources to move the vehicle
         ."},
        {searchterm: "aerodynamic drag", dbpedialabel: "Aerodynamic
           drag", dbpediauri: "http://dbpedia.org/resource/
           Aerodynamic_drag", dbpediadescription: "Aerodynamic drag
           is the fluid drag force that acts on any moving solid
           body in the direction of the fluid freestream flow."}
    ]
}
```

LOV querying method

Receives a list of terms and returns the LOV properties related to those.

Request content:

```
{
  keywords: [
      {word: "altitude", score: 45},
      {word: "coordinates", score: 34}
  ]
}
```

Response content:

```
{
  result: [
      {searchterm: "altitude", lovvocabulary: "dbpedia-owl", lovuri:
         "http://dbpedia.org/ontology/altitude", lovlabel:"altitude"
         },
        {searchterm: "coordinates", lovvocabulary: "gndo", lovuri:"
           http://d-nb.info/standards/elementset/gnd#coordinates",
           lovlabel:"coordinates"}
```

```
5       ]
6   }
```

5.3.2 User interface

Based on the previous API methods and the need to offer the curator/researcher a visual representation of the process we have developed an interface that was placed in the administration page of the project, meaning that only users with permissions may access it. It includes five different screens, each related to a step of the process: file selection, term and concept extraction, clustering and property selection.

The first, which can be seen in Fig 5.7, offers an overview of the files available in the project. In this area the user may choose the files that will be sent for preprocessing and term extraction.

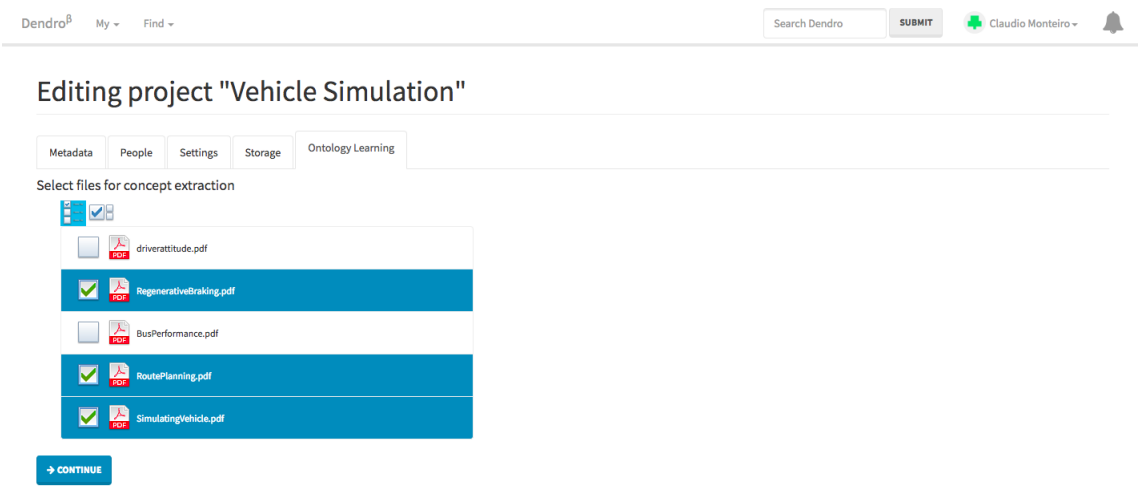


Figure 5.7: List of files within the project

After pressing the "Continue" button, the user is taken to the next step, in which, every extracted term is ranked by score, however this score is hidden from the user. Like the previous step, the user may now select the terms to be used in the DBpedia query. The selection may range from one to every term available. An example can be seen in Fig 5.8.

Figure 5.8: List of extracted terms ordered by score

The user will probably not find every wanted term in the previous list, so we have added an input box, which can be seen in Fig 5.9. This box allows the user to add terms that did not appear in the list, but he may consider either important or has doubts and wants to find more information about them. These added terms will then be added to the top of the term list and are automatically selected for the next step.

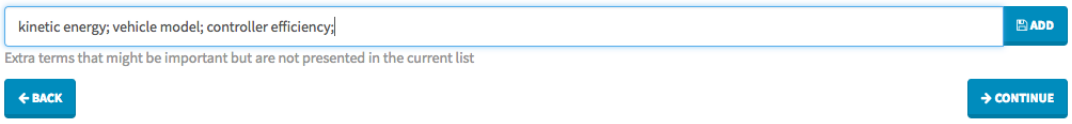


Figure 5.9: Input box for the addition of new search terms

The Cluster interface can be seen in Fig. 5.10 and allows the user to see the terms aggregated by lexical similarity as opposed to the extraction method score like as in Fig. 5.8. In this interface the user may select terms in two ways: either by pressing them individually or by pressing the whole box.

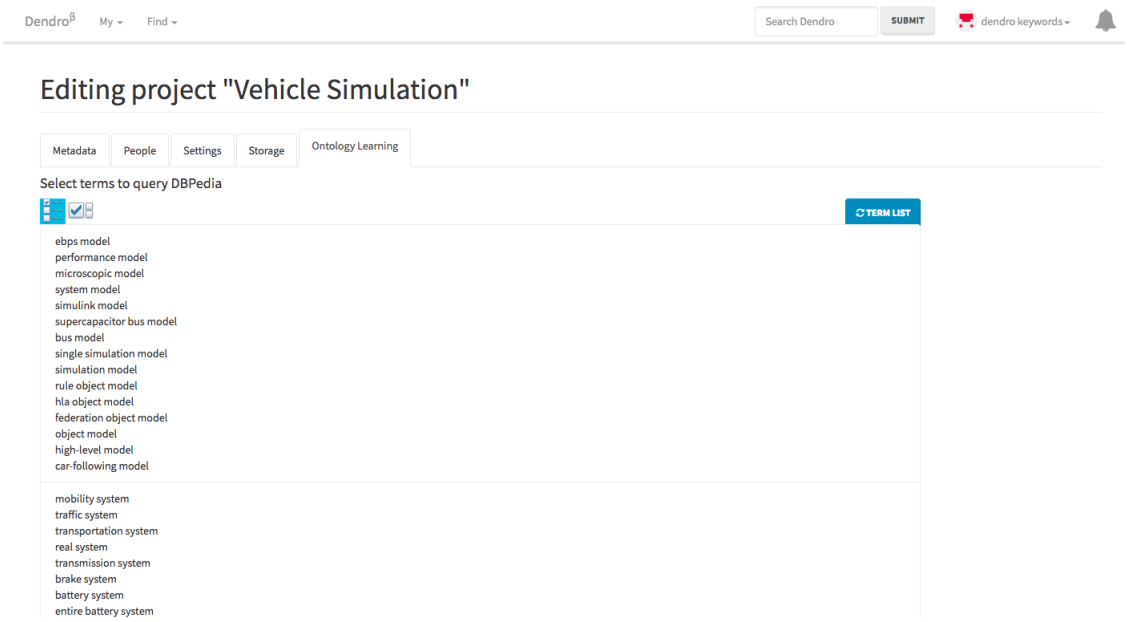


Figure 5.10: Example of clustering in Vehicle Simulation

After pressing "Continue" in the term extraction interface, DBpedia is consulted with the previously selected terms and then the user is taken to the output of this query, which can be seen in Fig 5.11. As expected we cannot always find a definition for a term in DBpedia, since in some cases the term used during the search is too specific. These concepts allow the curator, which is typically not an expert in the scientific domain of the data being described, to have an idea of what most of these domain terms mean and with that make a decision on their importance in the context of the domain. Based on the output and like the previous steps, the curator can selected which terms/concepts will have a descriptor presented.

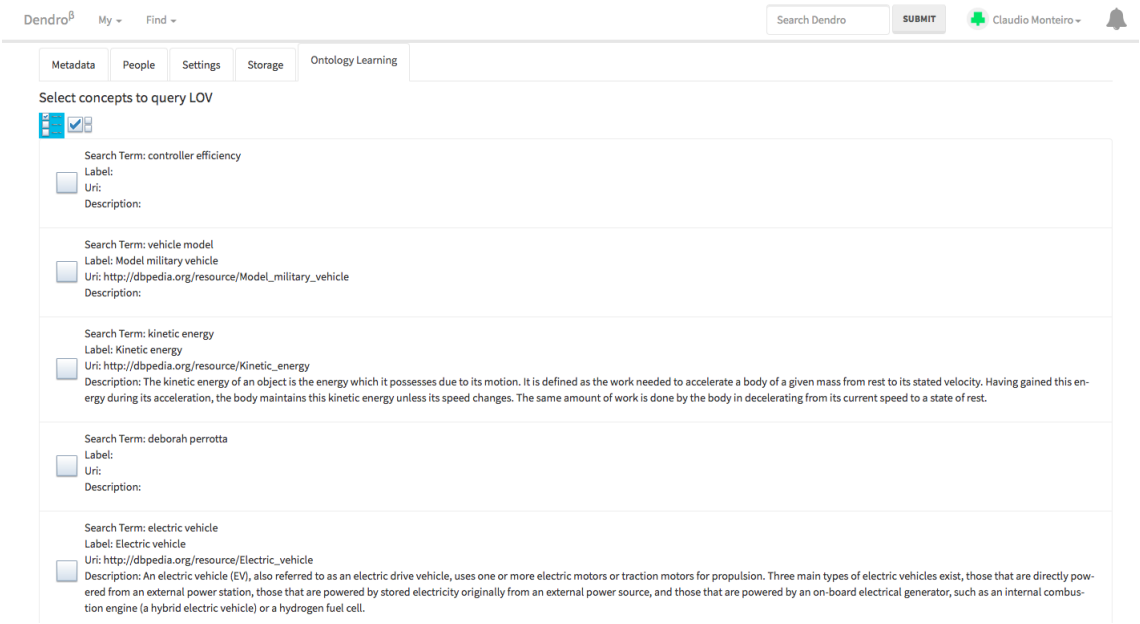


Figure 5.11: DBpedia Label, uri and description for the terms selected

The last view can be seen in Fig 5.12. This interface shows an overview of the descriptor proposed by LOV for the terms used. These results also contain the vocabulary and the URI for it in case the curator wants to know more about it. Even if these descriptors are considered to be too generic, they may still be used as a starting point to create a more specific descriptor that fits the domain being analyzed.

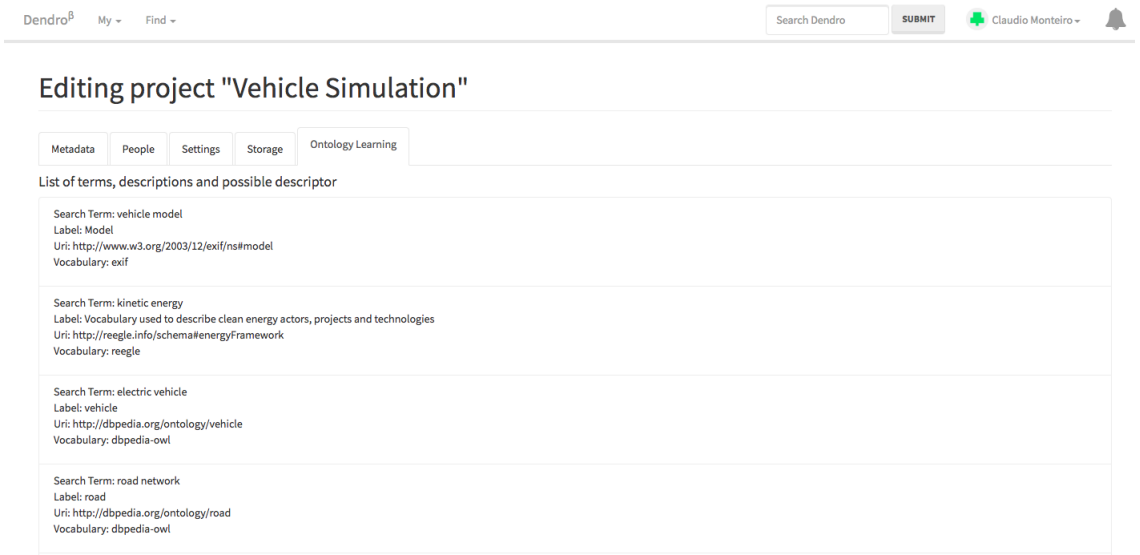


Figure 5.12: LOV properties based on the search terms

5.4 Summary

In this chapter we provided an overview of the methods used and their relation in the tool developed. We showed how Dendro Keywords should work when used either as an API or from within the project administration page. We have also shown examples for each task, instantiating the workflow of Dendro Keywords.

Chapter 6

Evaluation

6.1 Introduction

We have decided to evaluate Dendro Keywords in two ways. First we compared the output of our tool against the descriptors from the ontologies that were developed by the curators working on Dendro. These descriptors were also validated by researchers working on those specific domains. Then we conducted a user study with the same curators who were asked to use Dendro Keywords.

6.2 Evaluation scenario

Currently, there is not a general consensus on how an ontology should be evaluated and therefore the evaluation of an automatically generated ontology is a difficult task [Dry09].

In our case we will focus on two evaluation approaches [DS08]: automatic comparison with gold standard ontologies and manual evaluation. A gold standard is usually the best possible test for certain conditions and in our case a gold standard is an ontology that contains descriptors that have been previously validated by a domain expert.

In a manual evaluation the results are presented to a human expert, who will then judge them based on his knowledge. This approach has the possible disadvantage of being both time consuming or not having experts available.

6.2.1 Gold standard based evaluation

The metrics typically used to compare a created ontology with a gold standard are usually precision and recall as it was used in Chapter 4, yet there are systems that during their evaluation either used just precision [Dry09] or recall [Kva07].

In our case we have decided to analyze both precision and recall considering two comparison scenarios: 1. exact match between our result and a descriptor from the gold standard ontology and 2.‘if either our result or the descriptor contain each other (e.g. the ground-truth contains “Vehicle" but we only return “electric vehicle").

We decided to use the second comparison approach to cope with two issues: one is the descriptor possibly being too specific which would end in a lower number of matches and the other is that we only extract multi-worded terms causing descriptors which contain only one word to never find a match. As an example we have a descriptor on one of our gold standard ontologies named "vehicle", using C-Value we would not be able to find an exact match for it, but we were able to return similar terms such as "hybrid vehicle" or "electric vehicle". Using this approach we would now have a match.

For our gold standard evaluation we have used ontologies from three different domains from two curators that currently work on Dendro. From one of the curators a Vehicle Simulation ontology [CPA+15] and from the other a Sustainable Chemistry and Photovoltaic Application ontologies. In order to extract concepts from our tool to match against the gold standard we have used the same documents they used when manually creating those ontologies.

Also, in the case of the last two ontologies we have used documents provided by the researchers in the field that, for time constraints, were not considered by the curators to see how much the output would change. In total we have used 5 files for Vehicle Simulation, 3 and 13 for Photovoltaic Application and lastly 3 and 16 for Sustainable Chemistry. The reduced number of available papers add to the complexity of the task in hand, but only serves to highlight the hard task of the curators who face these situations everyday, and the importance of the proposed solution in assisting their work. Thus, we decided it would be good to do our evaluation using the same materials as the people that will benefit the most from this tool.

We have built Precision-recall curves for both exact match and partial match for the output of C-Value using the Noun+Noun and (Adj|Noun)+Noun filters and Yake!. We also created curves for the DBpedia and LOV results when using the terms extracted with those methods as query parameters. In order to simply the references to the C-Value filters we have decided to use the term CValueNN for the Noun+Noun filter and CValueJJ for the (Adj|Noun)+Noun filter. NN are the initials of the words in the filter, while JJ is related to the POS-tag of an adjective. We should also take into account that the values of X and Y in these curves may vary.

After, we did a comparison similar to the one made in Chapter 4, for which we calculated both precision and recall for the top terms extracted in each method.

Vehicle simulation

The Vehicle Simulation ontology contains a total of 12 descriptors [CPA+15]. The results for this first case can seen in Fig. 6.1. The first row shows the results related to the extraction of terms with the terms tested in Section 4, Yake! retrieved a total of 150 terms, or 125 when considering duplicate removal, while having no term with an exact match and only 2 matches where the terms searched either contain or are contained in the results ("contain" matches). CValueNN retrieved a total of 281 terms with 8 having an exact match and 10 in contain. CValueJJ retrieved 398 and had an exact match on 10 terms and 11 on the contain approach reaching a recall of 92%.

On the second row we have DBpedia, again we have no exact match using Yake! and got 3 and 4 exact matches with CValueNN and CValueJJ respectively. Using contain we have 1 with Yake!, 6 and then 7 reaching a maximum of 58% in recall.

Using LOV and considering an exact match every method got 1 match, which was the descriptor "vehicle". On contain, Yake! got 4 matches and the C-Value filters got 6, resulting in a 50% recall.

Looking at these curves, we can see that Yake! shows very low results, while the (Adj|Noun)+Noun filter of C-Value allowed us to find 92% of the descriptors. Regardless of the methods used, our precision hovered around 4%.

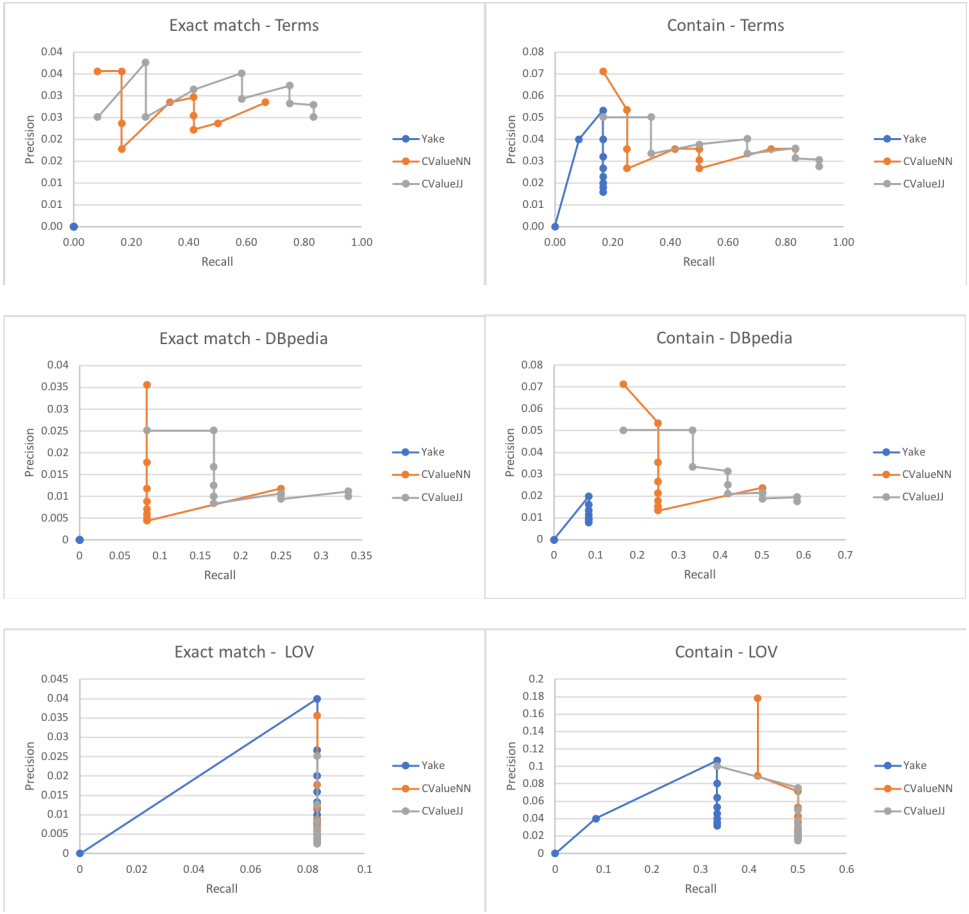


Figure 6.1: Precision vs recall graphs for Vehicle Simulation

Sustainable Chemistry

This ontology contains a total of 52 descriptors. We started by doing our evaluation using only the same 3 files as the curator.

The results for these tests can be seen in Fig. 6.2. Using exact match we were only able to find 1 exact match with Yake! and it was during the comparison with extracted terms, using the other methods we were able to find a maximum of 6 with CValueJJ. In this case we had no exact match when using LOV and this has to do with the fact that the available descriptors in LOV are

very generic [MLS18], while here, we are working with more specific terms, which will result in poor exact match in most cases.

When using contain we were able to find more matches, Yake! improved in every situation, with the best results when comparing with the terms, having a match on 16, which results in around 31% of all descriptors. As expected the best results were shown by CValueJJ with a match on 22 descriptors, which amounts to 42%. The LOV results came close with CValueNN filter reaching 40% of the results and the CValueJJ 38%.

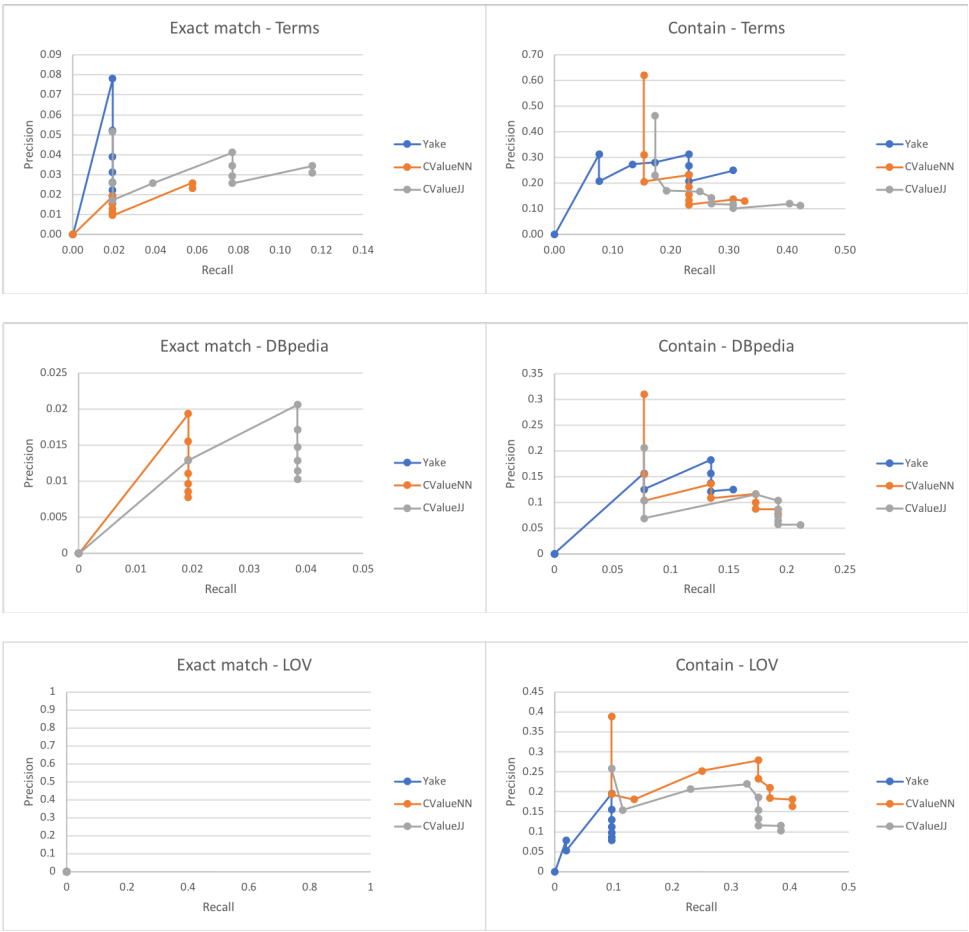


Figure 6.2: Precision vs recall graphs for Sustainable Chemistry using 3 files

We decided to use a bigger sample, containing 16 documents in order to see how much the results could change by using more documents during the term extraction phase. These results can be seen in Fig. 6.3. As expected, the precision decreased due to the much bigger amount of extracted terms (the number of terms extracted with CValueNN increased from 129 to 888, Yake! from 64 to 322 and CValueJJ from 194 to 1450).

In this case we also did not find any exact match with LOV descriptors. Considering DBpedia, the results were also poor, with Yake! not having a single match, followed by CValueNN with 1 and CValueJJ with 2.

Moving from 3 to 16 files, we found the biggest improvements were when comparing terms and LOV properties using the contain approach, we were able to see an increase of around 20% in matches.

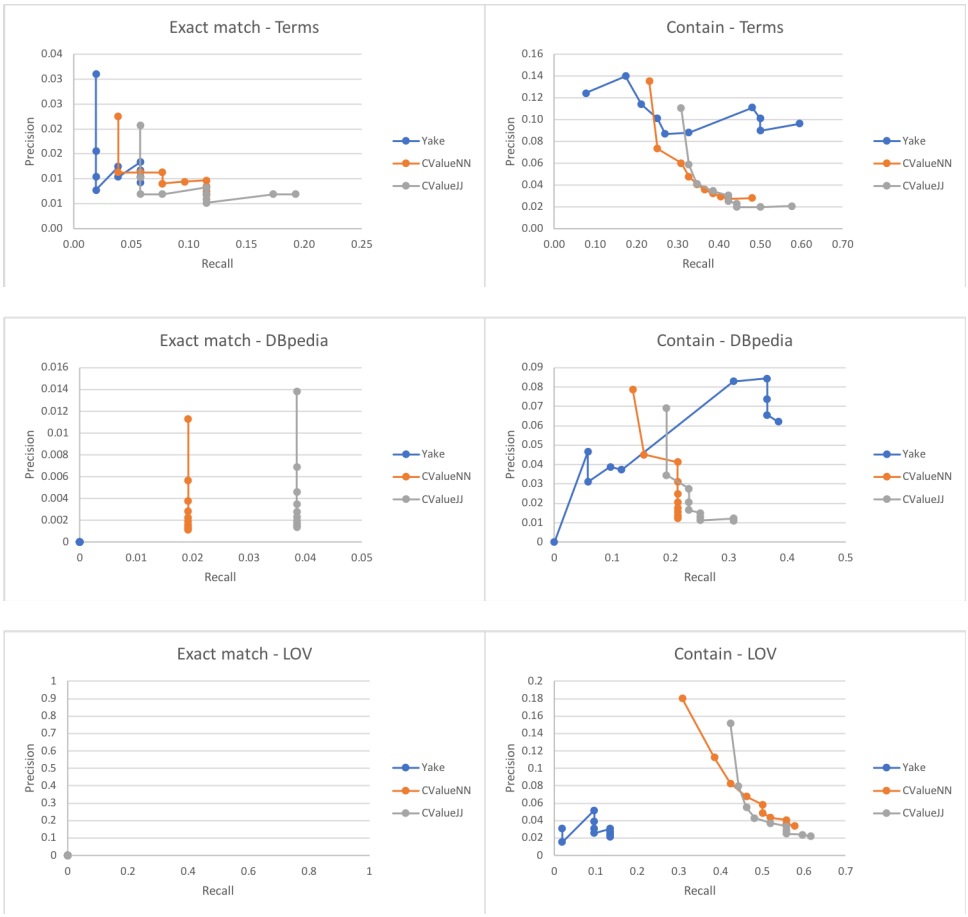


Figure 6.3: Precision vs recall graphs for Sustainable Chemistry using 16 files

Photovoltaic Application

The Photovoltaic Application is the last of the 3 domains being analyzed and its ontology contains 44 descriptors. This ontology followed the same manual creation methods as the previous, being built based on 3 documents. Our results based on these documents can be seen in Fig. 6.4.

Again, the results for Yake! were somewhat poor, apart from the contain comparison with the terms where it got 8 matches (18%), it was only able to find 1 exact match, also when using terms and 1 match with DBpedia when using contain.

Regarding the C-Value filters, CValueJJ was able to provide better results, getting a recall of 41% against the 32% from CValueNN. It was also the only method that was able to get an exact match when using LOV descriptors. Regarding the other approaches CValueJJ and CValueNN results were similar, but the first always offered extra matches.

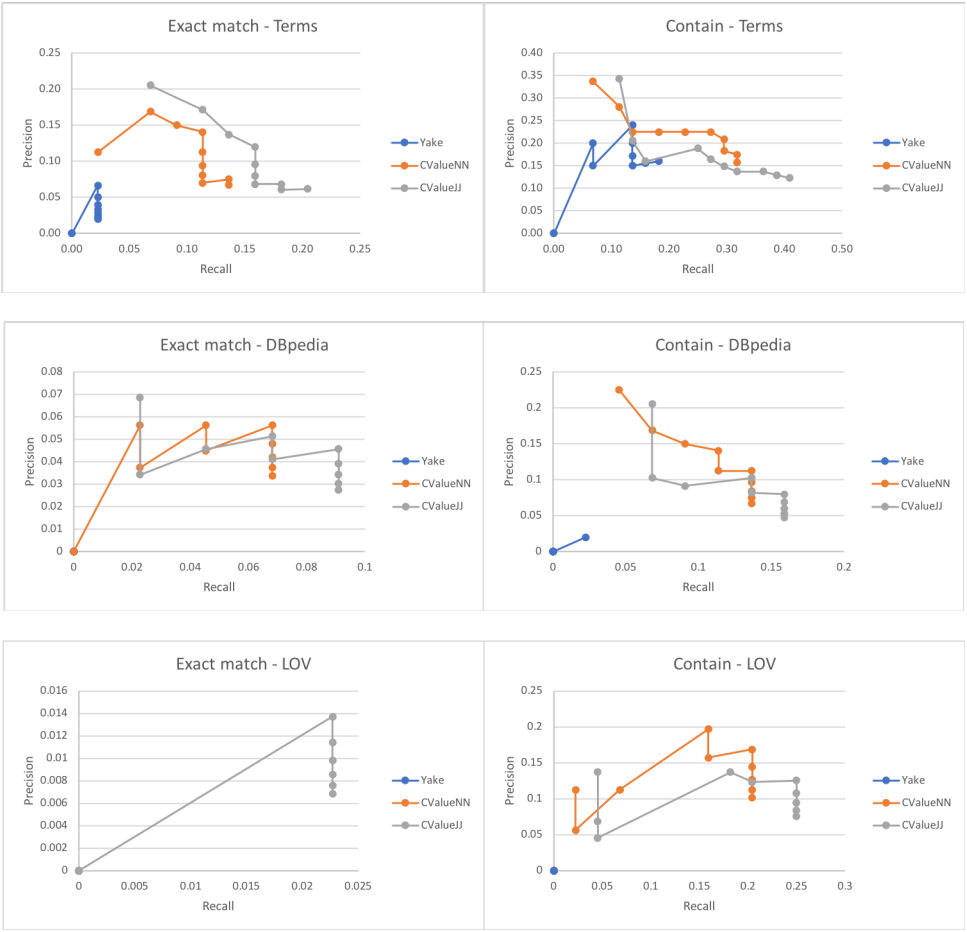


Figure 6.4: Precision vs recall graphs for Photovoltaic Application when using 3 files

Similar to what happened in the Sustainable Chemistry case, we are able to observe a decrease in precision, while also observing an increase in recall due to the increase in the number of files used in the evaluation and the terms extracted.

The results of the Photovoltaic Application case with 13 files can be seen in Fig. 6.5.

Surprisingly, we were able to observe Yake! providing better results than CValueJJ, even if only slightly. When using an exact match between the descriptor and the Yake! terms we were only able to find a match in 3 terms, while CValueNN had 9 and CValueJJ had 11. Yet, when using contain, Yake! obtained a match in 21, against 18 and 21 by the others. This resulted in a recall value of 42%. Using contain, Yake! was also able to match the results of CValueNN when using DBpedia.

The best results in this case were obtained when using C-Value. CValueNN reached a recall of 55% and CValueJJ 61%.

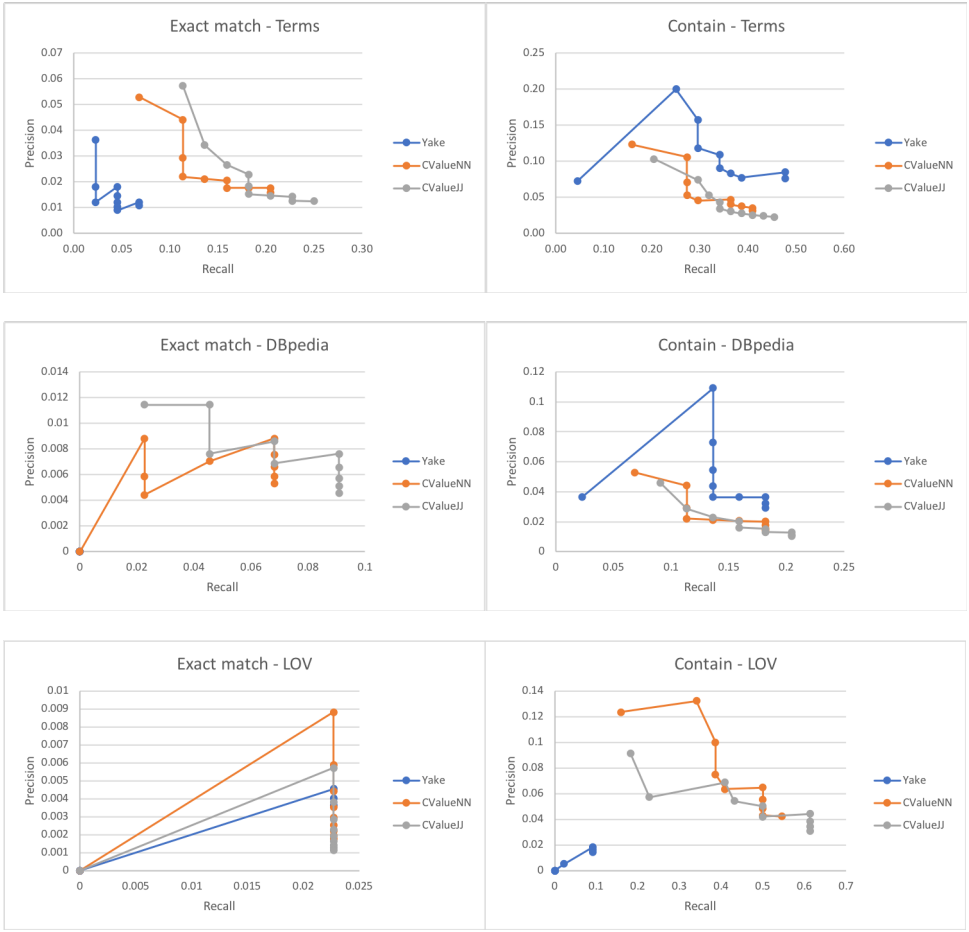


Figure 6.5: Precision vs recall graphs for Photovoltaic Application when using 13 files

Method comparison using top terms

The previous evaluation was done using every possible candidate term, so for this case, and as seen in the evaluation present in Chapter 4, we have decided to only use the top terms extracted. By using the same quantity of terms in each case we allow a more fair evaluation since the methods will be evaluated using the same conditions, unlike the previous case where C-Value extracted a much bigger quantity of terms than Yake!. We calculated precision and recall for the top 5,10,15 and 30 results.

For Vehicle Simulation the results can be seen in Table 6.1. When using the top terms extracted in each method we were only able to obtain 1 exact match and it was when we extracted the top 30 terms with each linguistic filter of C-Value. Using the contain approach we were also only able to extract 1 term with Yake!. The best scores we reached were 20% precision using the top 5 terms extracted and a maximum of 25% recall, which is equal to 3 terms extracted in 30.

	Top 5		Top 10		Top 15		Top 30	
	P	R	P	R	P	R	P	R
Yake! exact match	0	0	0	0	0	0	0	0
Yake! contain match	0	0	0	0	0	0	3.33	8.33
C-ValueNN exact match	0	0	0	0	0	0	3.33	8.33
C-ValueNN contain match	20	8.33	10	8.33	6.67	8.33	6.67	16.67
C-ValueJJ exact match	0	0	0	0	0	0	3.33	8.33
C-ValueJJ contain match	20	8.33	10	8.33	6.67	8.33	10	25

Table 6.1: Top terms extracted for Vehicle Simulation

The results for Sustainable Chemistry when using 3 files can be seen in Table 6.2. Using the exact match we were only able to find 1 result and it was when using C-ValueJJ. When using contain we were able to reach a precision of 100% for the top 5 terms using both C-Value filters, as for the top 30 terms, the precision was 30% and recall 17.31% which is equal to 9 terms out of 30.

	Top 5		Top 10		Top 15		Top 30	
	P	R	P	R	P	R	P	R
Yake! exact	0	0	0	0	0	0	0	0
Yake! contain	0	0	10	1.92	13.33	3.85	6.67	3.85
C-ValueNN exact	0	0	0	0	0	0	0	0
C-ValueNN contain	100	9.62	80	15.38	53.33	15.38	26.67	15.38
C-ValueJJ exact	0	0	0	0	0	0	3.33	1.92
C-ValueJJ contain	100	9.62	80	15.38	53.33	15.38	30	17.31

Table 6.2: Top terms extracted for Sustainable chemistry when using 3 files

The results for Sustainable Chemistry when using 16 files can be seen in Table 6.3. In this case we were able to extract 1 term for every method when using an exact match. When using contain, our best precision was reached when extracting the top 15 terms, for which 8 had a match and as for recall the best result was 17.31%, obtained when extracting the top 30 terms.

	Top 5		Top 10		Top 15		Top 30	
	P	R	P	R	P	R	P	R
Yake! exact	0	0	0	0	0	0	3.33	1.92
Yake! contain	0	0	0	0	0	0	13.33	7.69
C-ValueNN exact	20	1.92	10	1.92	6.67	1.92	3.33	1.92
C-ValueNN contain	20	1.92	40	7.69	53.33	15.38	30	17.31
C-ValueJJ exact	20	1.92	10	1.92	6.67	1.92	3.33	1.92
C-ValueJJ contain	20	1.92	40	7.69	53.33	15.38	30	17.31

Table 6.3: Top terms extracted for Sustainable chemistry when using 16 files

The results for Photovoltaic Application when using 3 files can be seen in Table 6.4. When using an exact match we were able to reach a maximum of 20% precision when using the top 15 terms and 11.36% recall when using the top 30. When using contain, our best values were also in this conditions but now the precision was 33.33% and recall 13.64 %.

	Top 5		Top 10		Top 15		Top 30	
	P		P	R	P	R	P	R
Yake! exact	0	0	0	0	6.67	2.27	3.33	2.27
Yake! contain	0	0	0	0	20	6.82	20	13.64
C-ValueNN exact	0	0	10	2.27	20	6.82	13.33	9.09
C-ValueNN contain	20	2.27	30	6.82	33.33	11.36	20	13.64
C-ValueJJ exact	0	0	20	4.55	20	6.82	16.67	11.36
C-ValueJJ contain	20	2.27	30	6.82	33.33	11.36	20	13.64

Table 6.4: Top terms extracted for Photovoltaic Application when using 3 files

The results for Photovoltaic Application when using 13 files can be seen in Table 6.5. Apart from Yake! which had its recall drop from 13.64% to 4.55% in the final comparison, the results were similar to the ones obtained in the previous case. The biggest changes were the extraction of one more relevant term when using the top 5 terms, which allowed the precision to reach 40% when using contain and the increase of the maximum recall in the final comparison from 13.64 to 15.91%.

	Top 5		Top 10		Top 15		Top 30	
	P		P	R	P	R	P	R
Yake! exact	0	0	0	0	6.67	2.27	3.33	2.27
Yake! contain	0	0	0	0	13.33	4.55	6.67	4.55
C-ValueNN exact	20	2.27	10	2.27	13.33	4.55	6.67	4.55
C-ValueNN contain	40	4.55	20	4.55	26.67	9.09	20	13.64
C-ValueJJ exact	20	2.27	10	2.27	13.33	4.55	10	6.82
C-ValueJJ contain	40	4.55	20	4.55	20	6.82	23.33	15.91

Table 6.5: Top terms extracted for Photovoltaic Application when using 13 files

6.2.2 Manual evaluation

The objective of this tool is not to provide a finished ontology but actually to help the curator during that process. It helps curators find concepts that may help them reach the final set of descriptors before validating with the domain expert. Fig. 6.6 shows a concept map built by the creator of the Vehicle Simulation ontology during the creation process. We can see that the map contains more concepts than final descriptors (12), which puts the results of this work into perspective and highlights the complexity of the task at hand.

In this case the manual evaluation has two main objectives: to show that the tool provides concepts useful for the creator during the ontology development process and also to show how it enhances their work, either by reducing the time they spent on this task, helping them find concepts they overlooked, among others.

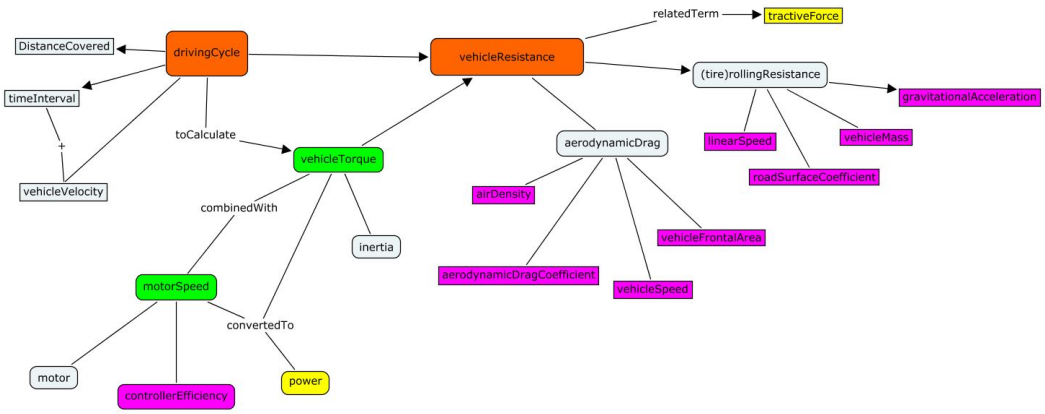


Figure 6.6: Map of concepts built by the curator during the creation of the Vehicle simulation ontology

Evaluation method

In order to do the manual evaluation, we have selected three different documents from the same domain [CKVO15, TP05, ASEH15] by using the query term “sugar" in Open Knowledge Maps[1]. We have only chosen three since the curators have said it is the usual amount of documents they analyze during the creation process. Using these documents, the curators prepared their own list of concept.

After their list was completed, we provided a walk through of the different phases of the tool. After they were familiarized with how the tool worked, we asked them to follow the different phases while answering to a questionnaire we previously prepared. This questionnaire had two questions they could answer before starting the process since these were related to their experience in the area. These questions were followed by questions regarding the different steps of the tool and, at the end, a section regarding possible relevant changes and how this tool could affect the way they work. In every question we used a 1 to 5 scale, 1 being strongly disagree and 5 strongly agree. An example of the questionnaire can be seen in Appendix B.

Questionnaire

We started by asking the curators about their experience with ontologies and also their experience with ontology learning tools. One of the curators selected very experienced, while the other selected average experience. Regarding ontology learning tools, none of them had any previous experience with these type of tools, apart from protégé [2] which is an ontology editor.

Curators then had to answer questions regarding usability. The overall process lasted 17 minutes for the first curator and 22 minutes for the second.

Fig. 6.7 shows the first questions, which are related to the term extraction phase. When asked if the presented terms were relevant for the domain, one curator said agree while the other said highly agree. When asked if the presented terms were the terms they expected from reading the documents beforehand both answered highly agree. In the end they were asked about the quantity and quality, both said the quantity was good, however one disagreed regarding the quality since although the curator stated that the terms were relevant for the domain, the curator was also expecting other type of terms, for example, "food" instead of different ingredients or "instrument" instead of the different instruments used.

[1] https://openknowledgemaps.org/
[2] https://protege.stanford.edu/

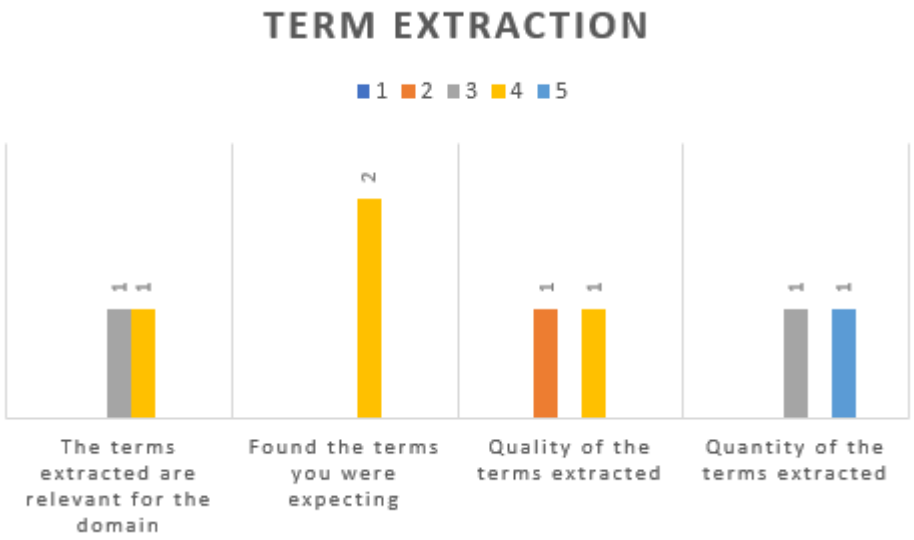


Figure 6.7: Questions related to term extraction

We then asked them to use clustering as a mean to visualize the different terms available and compare it to the list where terms were ordered by the extraction method score. One curator said the clustering helped while the other said it highly helped, but when asked which one they preferred both said the "Term List" interface. After being asked why, they stated that it was more visually appealing and since they had already used the normal list, it would not make sense to see the same terms again as groups.

We also asked them to rate both DBpedia concepts and LOV properties based on their quality and quantity. These results can be seen in Fig. 6.8. Both curators thought the quantity of the results provided was good, yet they were much more inclined to the properties provided by LOV, as expected since their main goal is to provide descriptors.

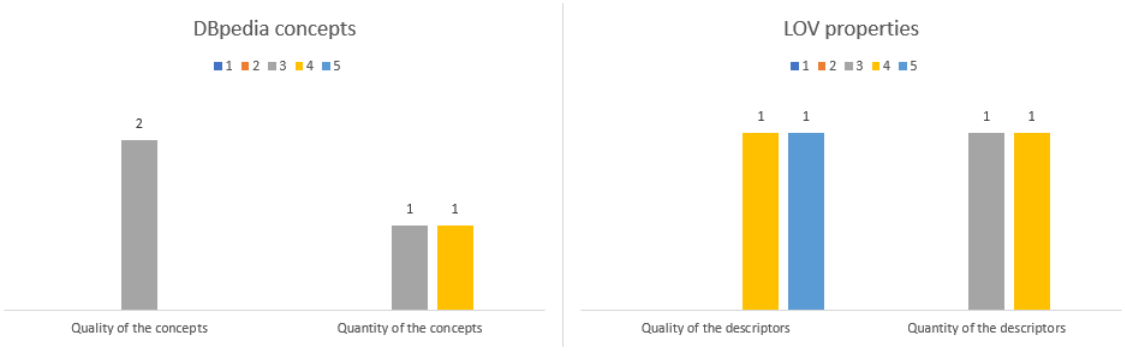


Figure 6.8: Questions related to DBpedia and LOV

After these questions, we asked them in which do they think this tool would be useful in their work. Both said that it would be very useful as a complement since it would allow the retrieval of concepts from multiple documents while the curator would only have to read a sample of them to

familiarize with the domain before speaking with the domain expert. By using the LOV feature it would also allow them to quickly check if concepts are already present in other ontologies.

Another question was which phases of the tool would the curator use more. Both curators said that the extraction of the LOV properties were the most important phase for them, while in second, one of the curators placed the term extraction and the other the DBpedia concepts. Both curators placed the lowest points in the clustering interface.

We followed this question by asking if the process was easy to achieve, and if it was not, what was the problem. The curator that answered that is was average, stated that the provided concepts were either too specific or incomplete and the way the process is currently done requires the curator to read about the context of the domain. The other curator although answering that the process was easy to achieve provided the same statement. The curator added that the interface of the tool is intuitive, however due to the nature of the extracted concepts, and without additional context, the curator would probably feel difficulties while defining the descriptors.

The last set of questions were related to possible relevant changes that could be implemented in each phase. Regarding the term extraction, curators would prefer less descriptors, while also being more generic, however they understand that for the extraction methods to work better it would require a much bigger sample of documents. They said clustering is interesting since it would allow the curator to have a much faster perception of the concepts available since the curator would not have to evaluate each one individually. Regarding DBpedia and LOV, they believe that terms without a match should be hidden instead of still appearing on the list. It would also be interesting for the curator to have feedback about the presence of similar descriptors to the wanted concept.

The last question of the questionnaire was about improvements to the overall process. Both of the curators answered similarly. They believe context should be extracted for each concept. For example, in a case where the curator finds a concept pertinent, he could select it and the tool would provide either one or multiple sentences containing that same concept in the corpus. This would provide sufficient knowledge for the curator to define a concept while not forcing him to read most of the domain documents.

6.3 Summary

In this chapter we provided an overview on some evaluation methods for ontology learning while also doing our own evaluation for the tool we proposed. When comparing with the Vehicle Simulation we were able to reach over 90% recall, while for the other cases we were able to pass the 60% mark. There are multiple reasons as to why our precision may be small, such as the number of documents used for testing, or the type of descriptors that the ontology contains. The ones available on Sustainable Chemistry and Photovoltaic Application were very specific most of of the times which we can observe in the gap in recall against vehicle simulation. Also, the work of the curator in these domains is based on deductions unlike in the first case, where the important concepts were explicit in the corpus. By reading the text if the curator finds, for example, 30°C,

the curator will know that it may be related to the descriptor "room temperature". The curators that did the manual evaluation of the tool were very satisfied with it and believe a tool like this would be a good addition to their workflow.

Chapter 7

Conclusions and Future Work

7.1 Summary

In this dissertation we have shown the importance of research data management and how metadata is at the center of data management. The process of creating an ontology can be very time and resource consuming and not every research team has a person specialized in these aspects. In order to tackle these problems, the main objective of this dissertation was to provide a tool that will help the curator and/or researcher during the ontology creation process. By evaluating different methods used in the ontology learning area we have developed Dendro Keywords, a module designed to assist data curators in the process of creating an ontology for data description in Dendro by providing the user with concepts and possible descriptors related with the documents of the project. We believe our system is a capable solution and will be a helpful addition that will allow users to work more efficiently.

In order to achieve this solution we started by evaluating different keyword extraction methods. We then decided to ally the methods with the best results with the power of DBpedia and LOV in order to draw information from some of the largest resource and vocabulary databases and use it in Dendro Keywords.

We decided to use both an automatic and a manual evaluation to assess the improvements introduced by the new Dendro module. Although the precision was not great, we were able to provide the user with a good quantity of concepts. The user study conducted done during the manual evaluation also showed how this work can be important in their workflow.

When compared to the tools described in Chapter 3 we offer some advantages. The use of DBpedia in order to find resources and the use of LOV to get a starting descriptor are some advantages. Our biggest advantage is providing a web based tool for ontology learning. A tool like text2onto not only possesses an outdated interface, but it also forces the user to install multiple software dependencies, such as a specific GATE or WordNet version.

In short, the proposed work was implemented and provides a good addition to researchers and curators.

7.2 Future work

Future work on this tool should start by experimenting with more recent techniques for term extraction. The latest ontology learning systems are starting to make use of more advanced machine learning techniques such as deep learning.

Another feature that we have in mind for future work is the addition of an option to download the output of the tool as an extension, such as, OWL in order to be able to continue the ontology development in other tools such as Protegé.

Currently, when the user closes the module the progress from the different tasks is lost. A good addition for the tool would be the option to save the current state. This option would not only allow the user to continue the build process later, but also be able to build on top of existing results if new documents are added to the project.

Although unlikely, a standalone application would be a good addition. It would allow the use of programming languages that have better support for information retrieval and machine learning tasks.

Integrate directly into Dendro as a way to help finding existing descriptors that are relevant to the description of each resource in the project, thus reducing the creation of duplicate descriptors.

Lastly, the addition of the suggestion made by the curators. The inclusion of context to the extracted terms would allow curators to save time by not having to read most of the domain documents since the given context would provide sufficient knowledge for the curator to define a concept.

References

[Abb08] Daisy Abbott. What is Digital Curation? *DCC Briefing Papers: Introduction to Curation*, (February), 2008.

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[ACdSR15] Ricardo Carvalho Amorim, João Aguiar Castro, João Rocha da Silva, and Cristina Ribeiro. A comparative study of platforms for research data management: interoperability, metadata capabilities and integration potential. *Advances in Intelligent Systems and Computing*, 353(February):III–IV, 2015.

[Alf10] Auhood Alfaries. Ontology Learning for Semantic Web Services. (September), 2010.

[ANS01] ANSI. The Dublin Core Metadata Element Set: standard ANSI. *Reading*, 2001.

[Arv02] Boulevard Pont-d Arve. Pre-processing very noisy text ISSCO / TIM University of Geneva Switzerland. pages 1–10, 2002.

[ASEH15] Heyam Ali, Rasha Saad, and Babiker El-Haj. Prevention of cap-locking of syrup product by treating the manufacturing process with citric acid monohydrate. *International journal of pharmaceutical chemistry*, 5(6):218–226, 2015.

[Bac08] Murtha Baca. *Introduction to metadata*. Getty Publications, 2008.

[Bar05] Phil Barker. What is IEEE Learning Object Metadata/IMS Learning Resource Metadata? *Cetis Standards Briefings Series*, 1(5):4, 2005.

[BCM05] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. *Ontology Learning from Text: An Overview*, volume 123, pages 3–12. IOS Press, 2005.

[Ben05] Amy Benson. *Metadata 101*. NELINET, Inc., 2005.

[BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227, 2009.

[Bie05] Chris Biemann. Ontology Learning from Text : A Survey of Methods. 20(2):75–93, 2005.

[BK11] Florian Bauer and Martin Kaltenböck. Linked open data: The essentials. *Edition mono/monochrom, Vienna*, 2011.

REFERENCES

[BKLB08] Steven Bird, Ewan Klein, Edward Loper, and Jason Baldridge. Multidisciplinary instruction with the Natural Language Toolkit. *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics - TeachCL '08*, (June):62, 2008.

[Bor03] Pia Borlund. The concept of relevance in information retrieval. *Journal of the American Society for Information Science and Technology*, 54(10):913–925, 2003.

[Bor10] C.L. Borgman. Research Data: Who will share what, with whom, when, and why? *China-North American Library Conference*, (September):21, 2010.

[Bor12] Christine L Borgman. The conundrum of sharing research data. *Journal of the American Society for Information Science and Technology*, 63(6):1059–1078, 2012.

[Bot15] Bothma Bothma. Ontology learning from swedish text, 2015.

[BSC99] Brigitte Biebow, Sylvie Szulman, and Av JB Clément. Terminae: A linguistics-based tool for the building of a domain ontology. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 49–66. Springer, 1999.

[Bur12] A. Burnham. Research Data - Definitions. *Research Data - Definitions*, pages 1–5, 2012.

[Can05] Linda Cantara. Mets: The metadata encoding and transmission standard. *Cataloging & classification quarterly*, 40(3-4):237–253, 2005.

[CASJ09] Philipp Cimiano, M Alexander, Steffen Staab, and V Johanna. Handbook on Ontologies. pages 245–267, 2009.

[CKA] About ckan. http://www.webcitation.org/6rJvkXWer. Accessed: 2017-06-11.

[CKVO15] Marcela Pedraza Carrillo, Samir Moura Kadri, Nabor Veiga, and Ricardo de Oliveira Orsi. Energetic feedings influence beeswax production by apis mellifera l. honeybees. *Acta Scientiarum. Animal Sciences*, 37(1):73–76, 2015.

[CMP+18a] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. A text feature based automatic keyword extraction method for single documents. In Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury, editors, *Advances in Information Retrieval*, pages 684–691, Cham, 2018. Springer International Publishing.

[CMP+18b] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. Yake! collection-independent automatic keyword extractor. In *European Conference on Information Retrieval*, pages 806–810. Springer, 2018.

[CPA+15] João Aguiar Castro, Deborah Perrotta, Ricardo Carvalho Amorim, João Rocha da Silva, and Cristina Ribeiro. Ontologies for research data description: a design process applied to vehicle simulation. In *Research Conference on Metadata and Semantics Research*, pages 348–354. Springer, 2015.

[Cun04] Morgan V Cundiff. An introduction to the metadata encoding and transmission standard (mets). *Library Hi Tech*, 22(1):52–64, 2004.

[CV05] Philipp Cimiano and Johanna Völker. Text2onto: A framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Natural Language Processing and Information Systems*, NLDB'05, pages 227–238, Berlin, Heidelberg, 2005. Springer-Verlag.

[Dar] The darwin core standard. http://www.webcitation.org/6rfFC7DZu. Accessed: 2017-06-6.

[DG08] Lucas Drumond and Rosario Girardi. A survey of ontology learning procedures. *CEUR Workshop Proceedings*, 427, 2008.

[DHSW02] Erik Duval, Wayne Hodgins, Stuart Sutton, and Stuart L Weibel. Metadata principles and practicalities. *D-lib Magazine*, 8(4):1082–9873, 2002.

[DJ86] Arthur M Diamond Jr. What is a citation worth? *Journal of Human Resources*, pages 200–215, 1986.

[Dry09] Efthymios G Drymonas. Ontology learning from text based on multi-word term concepts: The ontogain method. *Master of Science thesis, Technical University of Crete, Greece*, 2009.

[DS08] Klaas Dellschaft and Steffen Staab. Strategies for the evaluation of ontology learning. *Ontology Learning and Population*, 167:253–272, 2008.

[dSCRL14] João Rocha da Silva, João Aguiar Castro, Cristina Ribeiro, and João Correia Lopes. The dendro research data management platform. *IPRES 2014 proceedings*, page 189, 2014.

[DSM99] Jair Moura Duarte, João Bosco dos Santos, and Leonardo Cunha Melo. Comparison of similarity coefficients based on rapd markers in the common bean. *Genetics and Molecular Biology*, 22(3):427–432, 1999.

[DSp] Dspace, why use? http://www.webcitation.org/6rJuqkAVm. Accessed: 2017-06-11.

[dSRL14] João Rocha da Silva, Cristina Ribeiro, and João Correia Lopes. Ontology-based multi-domain metadata for research data management using triple stores. In *Proceedings of the 18th International Database Engineering & Applications Symposium*, pages 105–114. ACM, 2014.

[Dub03] Dublincore. Dublin Core Metadata Initiative. (16.06.2004), 2003.

[DV04] Hasan Davulcu and S Vadrevu. OntoMiner: bootstrapping ontologies from overlapping domain specific web sites. *international World Wide Web*, pages 500–501, 2004.

[DZP10] Euthymios Drymonas, Kalliopi Zervanou, and Euripides G. M. Petrakis. *Unsupervised Ontology Acquisition from Plain Texts: The OntoGain System*, pages 277–287. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

REFERENCES

[EW09] Nees Jan van Eck and Ludo Waltman. How to normalize cooccurrence data? an analysis of some well-known similarity measures. *Journal of the Association for Information Science and Technology*, 60(8):1635–1651, 2009.

[FAM00] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International journal on digital libraries*, 3(2):115–130, 2000.

[FGM06] Blaz Fortuna, Marko Grobelnik, and Dunja Mladenić. Semi-automatic data-driven ontology construction system. 2006.

[Fig] Figshare institutions and features. http://www.webcitation.org/6rJwAp0V5. Accessed: 2017-06-11.

[Fis87] Douglas H. Fisher. Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning*, 2(2):139–172, 1987.

[FMG06] Blaž Fortuna, Dunja Mladenič, and Marko Grobelnik. Semi-automatic construction of topic ontologies. *Semantics, Web and Mining*, pages 121–131, 2006.

[FN98] D. Faure and C. Nedellec. Asium: Learning Subcategorization Frames and Restrictions of Selection. *Proceedings of the 10th European Conference on Machine Learning, Workshop on Text Mining*, 409:410, 1998.

[FN99] David Faure and Claire Nédellec. *Knowledge acquisition of predicate argument structures from technical texts using Machine Learning: the system Asium*, pages 329–334. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[FNR] David Faure, Claire Nedellec, and Celine Rouveirol. Acquisition of Semantic Knowledge using Machine learning methods : The System " ASIUM " 1 Introduction. pages 1–19.

[fora] Programa 3 forum gdi. http://forumgdi.rcaap.pt/wp-content/uploads/2017/07/06-3ForumGDI_Dendro_Keywords.pdf. Accessed: 2018-06-25.

[Forb] Blaž Fortuna. Background Knowledge for Ontology Construction.

[Fur00] Betty Furrie. Understanding marc bibliographic: machine-readable cataloging. Cataloging Distribution Service, Library of Congress in collaboration with the Follett Software Company, 2000.

[G+16] Jiwei Guan et al. A study of the use of keyword and keyphrase extraction techniques for answering biomedical questions. 2016.

[Gar03] Richard Gartner. Mods: Metadata object description schema. *JISC Techwatch report TSW*, pages 03–06, 2003.

[GPR10] José Manuel Gómez-Pérez and Carlos Ruiz. Ontological engineering and the semantic web. In *Advanced Techniques in Web Intelligence-I*, pages 191–224. Springer, 2010.

[HBR09] Maryam Hazman, Samhaa R. El Beltagy, and Ahmed Rafea. Ontology learning from domain specific web documents. *International Journal of Metadata, Semantics and Ontologies*, 4(1/2):24, 2009.

[HCA05] Andrew Hippisley, David Cheng, and Khurshid Ahmad. The head-modifier principle and multilingual term extraction. *Natural Language Engineering*, 11(2):129–157, 2005.

[HHH+17] Sven Hartrumpf, Hermann Helbig, Fernuniversität Hagen, Sven Hartrumpf, Hermann Helbig, and Rainer Osswald. The semantically based computer lexicon HaGenLex The Semantically Based Computer Lexicon HaGenLex Structure and Technological Environment. (January 2003), 2017.

[Hig08] Sarah Higgins. The DCC Curation Lifecycle Model. *International Journal of Digital Curation*, 3(1):134–140, 2008.

[HR01] Udo Hahn and Martin Romacker. The syndikate text knowledge base generator. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–6, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics.

[HRR11] Maryam Hazman, Samhaa R. El-Beltagy, and Ahmed Rafea. A Survey of Ontology Learning Approaches. *International Journal of Computer Applications*, 22(9):36–43, 2011.

[Hwa99] CH Hwang. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. *Proceedings of the 6th International Workshop on . . .* , (Mcc):1–12, 1999.

[Ing16] C Ingram. How and why you should manage your research data: a guide for researchers. *An introduction to engaging with research data management processes, JISC, January*, 7, 2016.

[JA12] C. Jacinto and Claudia Antunes. User-Driven Ontology Learning from Structured Data. *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, pages 184–189, 2012.

[Jan88] Jane Greenberg. Metadata and the World Wide Web. *The University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, U.S.A.*, 25(4):215–227, 1988.

[JHC01] Istvan Jonyer, Lawrence B. Holder, and Diane J. Cook. Graph-Based Hierarchical Conceptual Clustering. *International Journal on Artificial Intelligence Tools*, 10(01n02):107–135, 2001.

[JT05] Xing Jiang and Ah-Hwee Tan. Mining ontological knowledge from domain-specific text documents. In *Data Mining, Fifth IEEE International Conference on*, pages 4–pp. Ieee, 2005.

[Kar12] Ievgen Karlin. An evaluation of nlp toolkits for information quality assessment, 2012.

[KMKB10] Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26. Association for Computational Linguistics, 2010.

[Kva07] Gøran Sveia Kvarv. Ontology learning-suggesting associations from text. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, 2007.

[Lar10] Ray R Larson. Introduction to information retrieval. *Journal of the American Society for Information Science and Technology*, 61(4):852–853, 2010.

[LFL98] Thomas K Landauer, Peter W Folt, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2):259–284, 1998.

[LHC11] Kaihong Liu, William R. Hogan, and Rebecca S. Crowley. Natural language processing methods and systems for biomedical ontology learning. *Journal of Biomedical Informatics*, 44(1):163 – 179, 2011. Ontologies for Clinical and Translational Research.

[Lin98] Dekang Lin. Automatic retrieval and clustering of similar words. *ACL '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 768–774, 1998.

[LLL08] Fei Liu, Feifan Liu, and Yang Liu. Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion. In *2008 IEEE Spoken Language Technology Workshop*, pages 181–184, Dec 2008.

[LP+04] Krister Lindén, Jussi Olavi Piitulainen, et al. Discovering synonyms and other related words. In *Proceedings of COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, 2004.

[Man93] Christopher D. Manning. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, ACL '93, pages 235–242, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics.

[Mar] Xiaowei Xu Martin Ester, Hans-Peter Kriegel, Jörg Sander. A Density Based Notion of Clusters in Large Spatial Databases with Noise.

[Med09] Olena Medelyan. *Human-competitive automatic topic indexing*. PhD thesis, The University of Waikato, 2009.

[Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[MLS18] Cláudio Monteiro, Carla Teixeira Lopes, and João Rocha Silva. Supporting description of research data : evaluation and comparison of term and concept extraction approaches. pages 1–4, 2018. [Accepted, to be published].

[MMV01] Alexander Maedche, Er Maedche, and Raphael Volz. The ontology extraction maintenance framework text-to-onto. In *In Proceedings of the ICDM'01 Workshop on Integrating Data Mining and Knowledge Management*, 2001.

[MS00] Alexander Maedche and Steffen Staab. Discovering conceptual relations from text. In *Proceedings of the 14th European conference on artificial intelligence*, pages 321–325. IOS Press, 2000.

[MW08] Olena Medelyan and Ian H Witten. Domain-independent automatic keyphrase indexing with small training sets. *Journal of the Association for Information Science and Technology*, 59(7):1026–1040, 2008.

[MXS05] Man Li, Xiao-Yong Du, and Shan Wang. Learning ontology from relational database. *2005 International Conference on Machine Learning and Cybernetics*, (August):3410–3415 Vol. 6, 2005.

[NK07] Thuy Dung Nguyen and Min-Yen Kan. Keyphrase extraction in scientific publications. In *International Conference on Asian Digital Libraries*, pages 317–326. Springer, 2007.

[NSA02] Goran Nenadić, Irena Spasić, and Sophia Ananiadou. Automatic discovery of term similarities using pattern mining. In *COLING-02 on COMPUTERM 2002: second international workshop on computational terminology-Volume 14*, pages 1–7. Association for Computational Linguistics, 2002.

[Ope] The 8 principles of government data. http://www.webcitation.org/6rfT76SCD. Accessed: 2017-07-02.

[PDF07] Heather A Piwowar, Roger S Day, and Douglas B Fridsma. Sharing detailed research data is associated with increased citation rate. *PloS one*, 2(3):e308, 2007.

[PDM10] Mark A Parsons, Ruth Duerr, and Jean-Bernard Minster. Data citation and peer review. *Eos, Transactions American Geophysical Union*, 91(34):297–298, 2010.

[PMR+14] Deborah Perrotta, José Luís Macedo, Rosaldo JF Rossetti, Jorge Freire de Sousa, Zafeiris Kokkinogenis, Bernardo Ribeiro, and João L Afonso. Route planning for electric buses: a case study in oporto. *Procedia-Social and Behavioral Sciences*, 111:1004–1014, 2014.

[Pow11] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[Pre04] NISO Press. Understanding metadata. *National Information Standards*, 20, 2004.

[Pro99] Medical Language Processing. A Semantic Lexicon for Medical Language Processing. 6(3):205–218, 1999.

[RdSACRCL14] João Rocha da Silva, João Aguiar Castro, Cristina Ribeiro, and João Correia Lopes. *Dendro: Collaborative Research Data Management Built on Linked Open Data*, pages 483–487. Springer International Publishing, Cham, 2014.

[RECC10] Stuart J. Rose, David W. Engel, Nicholas O. Cramer, and Wendy E. Cowley. Automatic keyword extraction from individual documents. 2010.

[REE03] Juan Ramos, Juramos Eden, and Rutgers Edu. Using TF-IDF to Determine Word Relevance in Document Queries. *Processing*, 2003.

[RH00] Manjula Patel Rachel Heery. Application Profiles: Mixing and Matching Metadata Schemas. *Ariadne*, 2000.

REFERENCES

[Rib14] Cristina Ribeiro. Management Using Triple Stores. pages 1–20, 2014.

[Ric44] Elizabeth M Richards. *Introduction to Cataloging and the Classification of Books*, volume 32. 1944.

[Rob77] Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

[Rob04] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

[RRL12] J Rocha, C Ribeiro, and J Correia Lopes. Managing multidisciplinary research data: Extending dspace to enable long-term preservation of tabular datasets. In *iPres 2012 Conference*, pages 105–108, 2012.

[RRL18] J. Rocha, C. Ribeiro, and J. Lopes. Ranking Dublin Core descriptor lists from user interactions: a case study with Dublin Core Terms using the Dendro platform. *International Journal on Digital Libraries*, Apr 2018.

[RSRF10] Eloy Rodrigues, Ricardo Saraiva, Cristina Ribeiro, and Eugénia Matos Fernandes. Os repositórios de dados científicos: estado da arte. page 54, 2010.

[RTMC05] Dave Reynolds, Carol Thompson, Jishnu Mukerji, and Derek Coleman. An assessment of RDF / OWLmodelling. *October*, 2(11):2005–189, 2005.

[Rys02] Jostein Ryssevik. The Data Documentation Initiative (DDI) metadata specification. *Ann Arbor, MI: Data Documentation Alliance.*, (March 2000), 2002.

[Sah05] Magnus Sahlgren. An introduction to random indexing. 2005.

[SBF98] Rudi Studer, V.Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–197, 1998.

[Sch94] Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, 1994.

[Sho10] Nian Shong. Pearson's versus Spearman's and Kendall's correlation coefficients for continuous data. *Graduate School of Public Health*, pages 1–53, 2010.

[Spe13] Read Chapter Speech. Part-of-speech tagging. (Chapter 12):1–55, 2013.

[SR15] Alisa Surkis and Kevin Read. Research data management. *Journal of Medical Library Association*, 10(3):154–156, 2015.

[SRC12] J Silva, C Ribeiro, and J Correia Lopes. Managing multidisciplinary research data: Extending DSpace to enable long-term preservation of tabular datasets, 2012.

[Sub10] L Venkata Subramaniam. Noisy text analytics. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page 5, 2010.

REFERENCES

[SVGK16] Scott Shaw, Andreas François Vermeulen, Ankur Gupta, and David Kjerrumgaard. Querying Semi-Structured Data. *Practical Hive*, pages 115–131, 2016.

[SWGM05] Marta Sabou, Chris Wroe, Carole Goble, and G. Mishne. Learning Domain Ontologies for Web Service Descriptions: an Experiment in Bioinformatics. *Proceedings of the 14th international conference on World Wide Web*, (Section 4):190–198, 2005.

[TP05] RB Thapa and S Pokhrel. Impact of supplement diets on flights of cross breed honeybee (apis mellifera l.). *Journal of the Institute of Agriculture and Animal Science*, 26:71–76, 2005.

[Tri] ontototext - what is rdf triplestore? http://www.webcitation.org/6rg58Afm9. Accessed: 2017-07-03.

[Tri13] Craig Trim. The art of tokenization. *IBM Developer Works*, 2013.

[TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Chap 8 : Cluster Analysis: Basic Concepts and Algorithms. *Introduction to Data Mining*, page Chapter 8, 2005.

[Tur01] Peter D Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Proceedings of the 12th European Conference on Machine Learning (ECML-2001), Freiburg, Germany*, pages 491–502, 2001.

[TW09] Nai-Lung Tsao and David Wible. A method for unsupervised broad-coverage lexical error detectioin and correction. *Proceedings of the NAACL HLT Workshop on Innovative Use of NLP for Building Educational Applications*, The NAACL(June):51–54, 2009.

[VAPVV17] Pierre-Yves Vandenbussche, Ghislain A Atemezing, María Poveda-Villalón, and Bernard Vatant. Linked open vocabularies (lov): a gateway to reusable semantic vocabularies on the web. *Semantic Web*, 8(3):437–452, 2017.

[VFN13] Paola Velardi, Stefano Faralli, and Roberto Navigli. OntoLearn Reloaded : A Graph-Based Algorithm for Taxonomy Induction. (October 2012), 2013.

[VHC07] Johanna Völker, Pascal Hitzler, and Philipp Cimiano. Acquisition of OWL DL axioms from lexical resources. *The Semantic Web: Research and Applications*, pages 670–685, 2007.

[VNCN05] Paola Velardi, Roberto Navigli, Alessandro Cuchiarelli, and R Neri. Evaluation of ontolearn, a methodology for automatic learning of domain ontologies. *Ontology Learning from Text: Methods, evaluation and applications*, 123:92, 2005.

[WBG+12] John Wieczorek, David Bloom, Robert Guralnick, Stan Blum, Markus D??ring, Renato Giovanni, Tim Robertson, and David Vieglais. Darwin core: An evolving community-developed biodiversity data standard. *PLoS ONE*, 7(1), 2012.

[Wha] Boston university- what is data. http://www.webcitation.org/6rJwSTg1I. Accessed: 2017-06-16.

REFERENCES

[WLB12] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text. *ACM Computing Surveys*, 44(4):1–36, 2012.

[Won09] Wy Wong. Learning lightweight ontologies from text across different domains using the web as background knowledge. (September):282, 2009.

[YGTH00] Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic Acquisition of Domain Knowledge for Information Extraction. *Proceedings of the 18th ICCL*, 2:940–946, 2000.

[Zen] Zenodo features. http://www.webcitation.org/query?id=1497818485869141. Accessed: 2017-06-11.

Appendix A

Term extraction example

The descriptors for the Vehicle Simulation ontology and an example of the output generated from the different term extraction methods is presented. The descriptors are present in Table A.1 and the different outputs for the top 10 terms extracted using each method can be seen in Table A.2.

Descriptors	
Aerodynamic drag coefficient	Gear ratio
Tire radius	Air density
Vehicle mass	Driving cycle
Vehicle model	Vehicle frontal area
Vehicle	Road surface coefficient
Controller efficiency	Gravitational acceleration

Table A.1: Vehicle Simulation descriptors

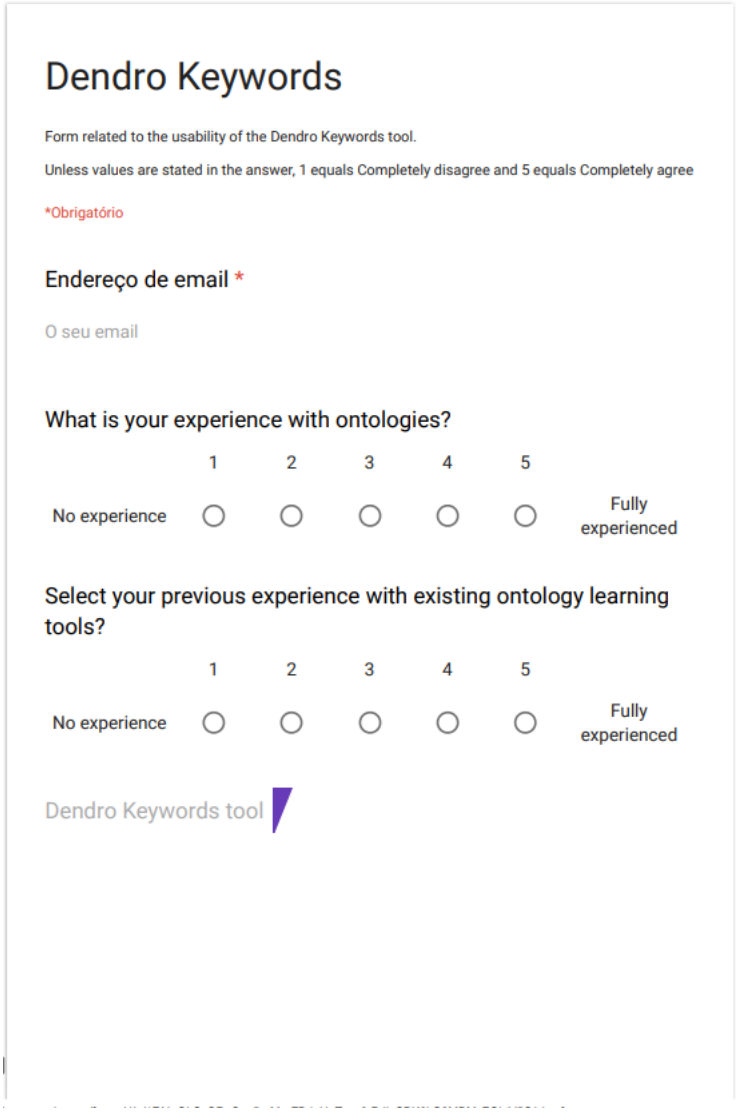
Terms extracted		
YAKE!	C-ValueNN	C-ValueJJ
porto	deborah perrotta	deborah perrotta
centro	electric vehicle	electric vehicle
euro working	kinetic energy	kinetic energy
working group	behavioral sciences	behavioral sciences
performance	bus stop	electric bus powertrain
deborah perrottaa	electric bus powertrain	bus stop
universidade	battery pack	battery pack
behavioral	bus powertrain	resistance force
authors	resistance force	bus powertrain
informatics engineering	energy consumption	energy consumption

Table A.2: Top 10 terms extracted for Vehicle Simulation in each method

Appendix B

Questionnaire

Here, the questionnaire made to curators is presented.



Dendro Keywords

Form related to the usability of the Dendro Keywords tool.

Unless values are stated in the answer, 1 equals Completely disagree and 5 equals Completely agree

*Obrigatório

Endereço de email *

O seu email

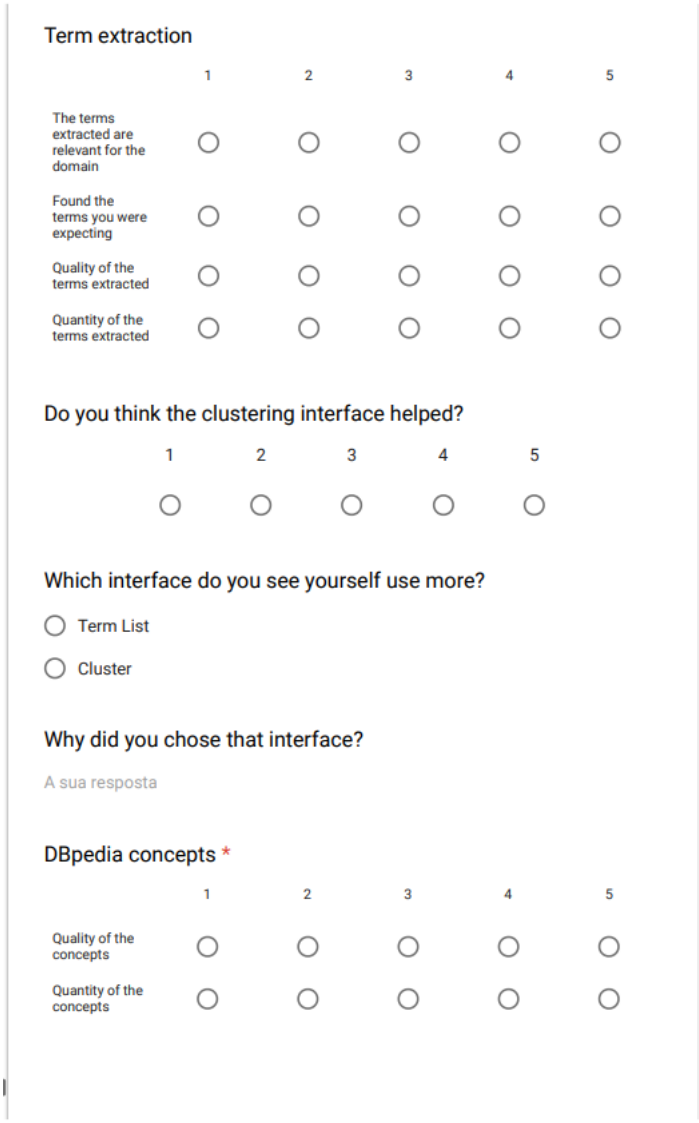
What is your experience with ontologies?

	1	2	3	4	5	
No experience	○	○	○	○	○	Fully experienced

Select your previous experience with existing ontology learning tools?

	1	2	3	4	5	
No experience	○	○	○	○	○	Fully experienced

Dendro Keywords tool



Term extraction

	1	2	3	4	5
The terms extracted are relevant for the domain	○	○	○	○	○
Found the terms you were expecting	○	○	○	○	○
Quality of the terms extracted	○	○	○	○	○
Quantity of the terms extracted	○	○	○	○	○

Do you think the clustering interface helped?

1	2	3	4	5
○	○	○	○	○

Which interface do you see yourself use more?

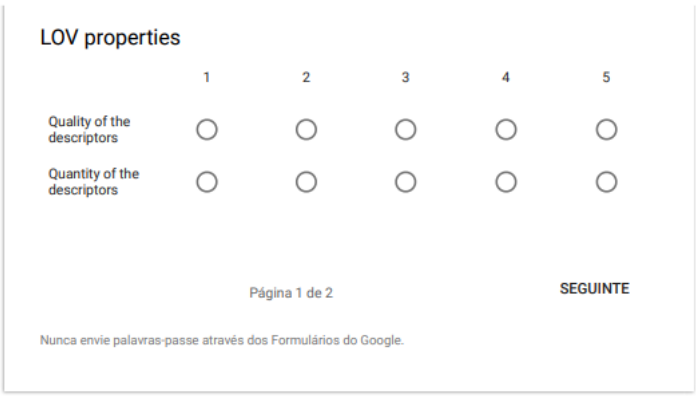
- ○ Term List
- ○ Cluster

Why did you chose that interface?

A sua resposta

DBpedia concepts *

	1	2	3	4	5
Quality of the concepts	○	○	○	○	○
Quantity of the concepts	○	○	○	○	○



LOV properties

	1	2	3	4	5
Quality of the descriptors	○	○	○	○	○
Quantity of the descriptors	○	○	○	○	○

Página 1 de 2 SEGUINTE

Nunca envie palavras-passe através dos Formulários do Google.

Dendro Keywords

Open answer section

In which way do you think this tool will be useful for your work?

A sua resposta

Which phases of the tool do you see yourself using more?

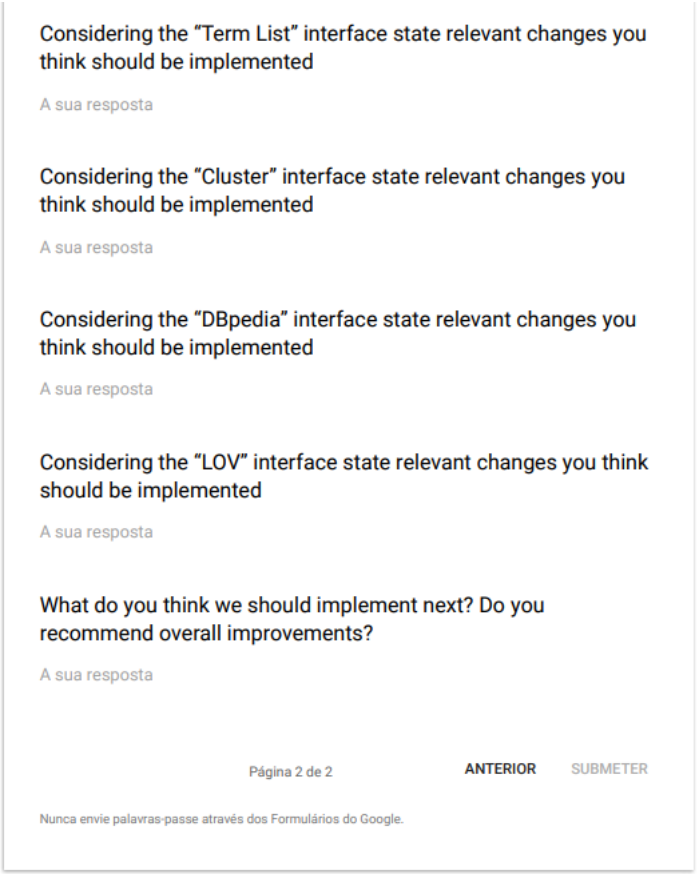
	1	2	3	4	5
Term Extraction	☐	☐	☐	☐	☐
Term Clustering	☐	☐	☐	☐	☐
DBpedia concepts	☐	☐	☐	☐	☐
LOV properties	☐	☐	☐	☐	☐

The end result was easy to achieve

1	2	3	4	5
○	○	○	○	○

If you disagreed in the previous answer, what was the problem?

A sua resposta



Considering the "Term List" interface state relevant changes you think should be implemented

A sua resposta

Considering the "Cluster" interface state relevant changes you think should be implemented

A sua resposta

Considering the "DBpedia" interface state relevant changes you think should be implemented

A sua resposta

Considering the "LOV" interface state relevant changes you think should be implemented

A sua resposta

What do you think we should implement next? Do you recommend overall improvements?

A sua resposta

Página 2 de 2 ANTERIOR SUBMETER

Nunca envie palavras-passe através dos Formulários do Google.